



(12) **EUROPEAN PATENT APPLICATION**  
published in accordance with Art. 158(3) EPC

(43) Date of publication:  
**26.02.2003 Bulletin 2003/09**

(51) Int Cl.7: **H04N 5/93, G11B 20/10**

(21) Application number: **01921965.8**

(86) International application number:  
**PCT/JP01/03416**

(22) Date of filing: **20.04.2001**

(87) International publication number:  
**WO 01/082609 (01.11.2001 Gazette 2001/44)**

(84) Designated Contracting States:  
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU**  
**MC NL PT SE TR**  
Designated Extension States:  
**AL LT LV MK RO SI**

(72) Inventors:  
• **HAMADA, Toshiya, SONY CORPORATION**  
**Shinagawa-ku, Tokyo 141-0001 (JP)**  
• **KATO, Motoki, SONY CORPORATION**  
**Shinagawa-ku, Tokyo 141-0001 (JP)**

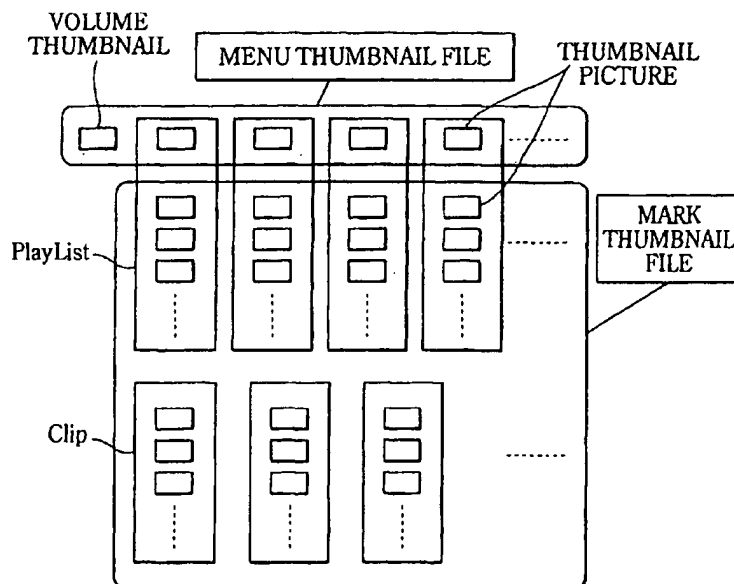
(30) Priority: **21.04.2000 JP 2000185479**

(74) Representative: **Pratt, Richard Wilson et al**  
**D. Young & Co,**  
**21 New Fetter Lane**  
**London EC4A 1DA (GB)**

(54) **RECORDING APPARATUS AND METHOD, REPRODUCING APPARATUS AND METHOD, RECORDED MEDIUM, AND PROGRAM**

(57) From input moving picture data, a thumbnail picture representing the moving picture data is generated as first thumbnail data. From the moving picture data, a thumbnail picture of a picture extracted as being a characteristic picture, or a thumbnail picture of a picture

specified by a user, is generated as second thumbnail data, and the first and second thumbnail data are recorded as respective independent groups on the recording medium. So, the user is allowed to select recorded data readily using the recorded picture data.



**FIG.20**

## Description

### Technical Field

[0001] This invention relates to a recording method and apparatus, a reproducing method and apparatus, a recording medium and a program, in which a thumbnail, representative of a picture, is appended to data.

### Background Art

[0002] Recently, a variety of types of optical discs have been proposed as a recording medium that can be removed from a recording apparatus. These recordable optical discs have been proposed as a large capacity medium of several GBs and are thought to be promising as a medium for recording AV (audio visual) signals, such as video signals. Among the sources (supply sources) of digital AV signal, to be recorded on this recordable optical disc, there are CS digital satellite broadcast and BS digital broadcast. Additionally, the ground wave television broadcast of the digital system has also been proposed for future use.

[0003] The digital video signals, supplied from these sources, are routinely compressed under the MPEG (Moving Picture Experts Group) 2 system. In a recording apparatus, a recording rate proper to the apparatus is set. If digital video signals of the digital broadcast are recorded in the conventional image storage mediums for domestic use, digital video signals are first decoded and subsequently bandwidth-limited for recording. In the case of the digital recording system, including, of course, the MPEG1 Video, MPEG2 video and DV system, digital video signals are first decoded and subsequently re-encoded in accordance with for the recording rate and the encoding system proper to the apparatus for subsequent recording.

[0004] However, this recording system, in which the supplied bitstream is decoded once and subsequently bandwidth-limited and re-encoded prior to recording, suffers from the deteriorated picture quality. If, in recording compressed digital signals, the transmission rate of input digital signals is less than the recording rate for the recording and/or reproducing apparatus, the method of directly recording the supplied bitstream without decoding or re-encoding suffers from deterioration in the picture quality to the least extent. However, if the transmission rate of the input digital signals exceeds the recording rate of the recording and/or reproducing apparatus, it is indeed necessary to re-encode the bitstream and to record the so re-encoded bitstream, so that, after decoding in the recording and/or reproducing apparatus, the transmission rate will be not higher than the upper limit of the disc recording rate.

[0005] If the bitstream is transmitted in a variable rate system in which the bit rate of the input digital signal is increased or decreased with time, the capacity of the recording medium can be exploited less wastefully with

a disc recording apparatus adapted for transiently storing data in a buffer and for recording the data in a burst fashion than with a tape recording system having a fixed recording rate imposed by the fixed rpm of the rotary head.

[0006] Thus, it may be predicted that, in the near future when the digital broadcast is to become the mainstream, an increasing demand will be raised for a recording and/or reproducing apparatus in which broadcast signals are recorded as digital signals, without decoding or re-encoding, as in a data streamer, and in which a disc is used as a recording medium.

[0007] In reproducing a recording medium, having recorded thereon plural data, such as data of program made up e.g., of video or speech data, using the above-described apparatus, there is presented a problem that, with an increasing volume of data recorded on the recording medium, the pre-playback processing, such as processing as to which program is to be reproduced or from which scene the program is to be reproduced, becomes complicated.

### Disclosure of the Invention

[0008] It is an object of the present invention to enable data for reproduction to be selected easily by adding a thumbnail representative of the data to the data to be reproduced.

[0009] A recording method according to the present invention a recording apparatus including means for generating, from input moving picture data, a thumbnail picture representing the moving picture data, as first thumbnail data, and for generating, from the moving picture data, a thumbnail picture of a picture extracted as a characteristic picture, or a thumbnail picture of a picture specified by a user, as a second thumbnail data, and

means for recording the first thumbnail data and the second thumbnail data, generated by the generating means as independent groups on a recording medium.

[0010] Preferably, the generating means generates the first and second thumbnail data as respective independent files.

[0011] Preferably, the generating means generates the first and second thumbnail data as respective independent data blocks and as one file.

[0012] Preferably, the generating means generates first management data supervising the first thumbnail data with a number corresponding to the first thumbnail data. The generating means also generates second management data supervising the second thumbnail data with a number corresponding to the second thumbnail data, whilst the recording means records the first and second management data in the recording medium.

[0013] Preferably, the first and second management data include data specifying the format of the picture data of the thumbnail data being supervised.

[0014] Preferably, the recording means records pic-

ture data of the thumbnail picture contained in the first or second management data in terms of a block of a preset size as a unit.

**[0015]** Preferably, the recording means records the information representing the referencing destination of the first thumbnail data as a separate file on the recording medium.

**[0016]** Preferably, the recording means further records the information indicating the referencing destination of the thumbnail picture contained in the second thumbnail data.

**[0017]** A recording method according to the present invention includes a generating step of generating, from input moving picture data, a thumbnail picture representing the moving picture data, as first thumbnail data, and for generating, from the moving picture data, a thumbnail picture of a picture extracted as a characteristic picture, or a thumbnail picture of a picture specified by a user, as a second thumbnail data, and a recording controlling step of exercising control for recording the first thumbnail data and the second thumbnail data, generated by the generating step as independent groups on a recording medium.

**[0018]** A program for a recording medium according to the present invention includes a generating step of generating, from input moving picture data, a thumbnail picture representing the moving picture data, as first thumbnail data, and for generating, from the moving picture data, a thumbnail picture of a picture extracted as a characteristic picture, or a thumbnail picture of a picture specified by a user, as a second thumbnail data, and

a recording controlling step of exercising control for recording the first thumbnail data and the second thumbnail data, generated by the generating step as independent groups on a recording medium.

**[0019]** A program according to the present invention has a computer execute a generating step of generating, from input moving picture data, a thumbnail picture representing the moving picture data, as first thumbnail data, and for generating, from the moving picture data, a thumbnail picture of a picture extracted as a characteristic picture, or a thumbnail picture of a picture specified by a user, as a second thumbnail data, and a recording controlling step of recording the first thumbnail data and the second thumbnail data, generated by the generating step as independent groups on a recording medium.

**[0020]** A reproducing apparatus according to the present invention includes first readout means for reading out management data supervising picture data of a thumbnail picture representing the contents of the picture data in case playback of the picture data is commanded, second readout means for reading out the picture data based on the management data read out by the readout means, third readout means for reading out picture data of a thumbnail picture of a picture extracted as a characteristic picture from the picture data or pic-

ture data of a thumbnail picture of a picture specified by a user, and management data supervising the picture data, and fourth readout means for reading out the picture data based on the management data read out from the third readout means.

**[0021]** A reproducing method according to the present invention includes a first readout controlling step of reading out management data supervising picture data of a thumbnail picture representing the contents of the picture data in case playback of the picture data is commanded, a second readout controlling step of controlling the readout of the picture data based on the management data readout-controlled by the processing of the first readout control means, a third readout controlling step of reading out picture data of a thumbnail picture of a picture extracted as a characteristic picture from the picture data or picture data of a thumbnail picture of a picture specified by a user, and management data supervising the picture data, and a fourth readout controlling step of reading out the picture data based on the management data readout-controlled by the processing of the third readout control step.

**[0022]** A program for a recording medium according to the present invention includes a first readout controlling step of controlling the readout of management data supervising picture data of a thumbnail picture representing the contents of the picture data in case playback of the picture data is commanded, a second readout controlling step of controlling the readout of the picture data based on the management data readout-controlled by the processing of the first readout control means, a third readout controlling step of reading out picture data of a thumbnail picture of a picture extracted as a characteristic picture from the picture data or picture data of a thumbnail picture of a picture specified by a user, and management data supervising the picture data, and a fourth readout controlling step of reading out the picture data based on the management data readout-controlled by the processing of the third readout controlling step.

**[0023]** A program according to the present invention includes a first readout controlling step of controlling the readout of management data supervising picture data of a thumbnail picture representing the contents of the picture data in case playback of the picture data is commanded, a second readout controlling step of controlling the readout of the picture data based on the management data readout-controlled by the processing of the first readout control means, a third readout controlling step of reading out picture data of a thumbnail picture of a picture extracted as a characteristic picture from the picture data or picture data of a thumbnail picture of a picture specified by a user, and management data supervising the picture data, and a fourth readout controlling step of reading out the picture data based on the management data readout-controlled by the processing of the third readout control step.

**[0024]** A recording medium according to the present invention has, recorded thereon, first data comprised of

picture data, thumbnail picture data of a thumbnail picture extracted from the picture data, indicating the contents of the picture data, and management data for supervising data of the thumbnail picture, and second data comprised of thumbnail picture data of a thumbnail picture of a picture extracted as a characteristic picture from the first stated picture data or picture data of a thumbnail picture of a picture specified by the user and management data supervising the picture data.

[0025] With the recording method and apparatus and the first program of the present invention, a thumbnail picture representative of input moving picture data is generated from the input moving picture data, as a first thumbnail data, whilst a thumbnail picture of a picture extracted as being a characteristic picture from the moving picture data or a thumbnail picture of a picture specified by the user is generated as second thumbnail data. The first and second thumbnail data are recorded as separate independent groups on the recording medium.

[0026] With the reproducing method and apparatus and the second program according to the present invention, if playback of an AV stream is commanded, management data supervising picture data of a thumbnail picture representing the contents of the AV stream is read out, whilst picture data is read out based on the management data read out by readout means. From the AV stream, picture data of a thumbnail picture of a picture extracted from it as a characteristic picture, or picture data of a thumbnail picture of a picture specified by a user, and management data supervising the picture data, are read out, and the picture data are read out based on the read-out management data.

[0027] Other objects, features and advantages of the present invention will become more apparent from reading the embodiments of the present invention as shown in the drawings.

#### Brief Description of the Drawings

[0028]

Fig.1 shows a configuration of a recording and/or reproducing apparatus according to the present invention.

Fig.2 shows the relation between Clip and PlayList.

Fig.3 is a UML doagram of a structure supervising an AV stream..

Fig.4 shows a directory structure of a DVR system..

Fig.5 shows info.dvr.

Fig.6 illustrates DVRVolume().

Fig.7 illustrates UIAppInfovolume().

Fig.8 illustrates TableOf PLayerList().

Fig.9 illustrates UIAppInfo PlayList().

Fig.10 illustrates zzzzz.clip.

Fig.11 illustrates ClipMark().

Fig.12 illustrates Mark.

Fig.13 illustrates xxxxx.rpls and yyyyy.vpls.

Fig.14 illustrates PlayList().

Fig.15 illustrates PlayItem().

Fig.16 illustrates PlayListMark().

Fig.17 illustrates a menu thumbnail.

Fig.18 illustrates a mark affixed to PlayList.

Fig.19 illustrates a mark affixed to the PlayList.

Fig.20 illustrates a file in which to store a thumbnail.

Fig.21 illustrates menu\_thmb/mark.thmb.

Fig.22 illustrates Thumbnail().

Fig.23 illustrates thumbnail\_picture\_format.

Figs.24A and 24B illustrate a method for storing thumbnail picture data in tn\_block().

Fig.25 illustrates another directory file structure.

Fig.26 illustrates menu\_thmb/mark.thmb associated with the file structure shown in Fig.25.

Fig.27 illustrates TableOfplayList() associated with the file structure shown in Fig.25.

Fig.28 illustrates PlayList() associated with the file structure shown in Fig.25.

Fig.29 illustrates UIAppInfo PlayList() associated with the file structure shown in Fig.25.

Fig.30 is a flowchart showing the sequence for creating a mark thumbnail.

Fig.31 is a flowchart showing the sequence for creating a menu thumbnail.

Fig.32 shows another directory structure of the DVR system.

Fig.33 shows the syntax of the thumbnail header information

Fig.34 illustrates display\_aspect\_ratio.

Fig.35 illustrates color\_space.

Fig.36 shows the syntax of a thumbnail picture data file.

Fig.37 illustrates data storage to tn\_block.

Fig.38 illustrates a medium.

#### Best Mode for Carrying out the Invention

[0029] Referring to the drawings, a recording method and apparatus, a reproducing method and apparatus, a recording medium and a program, according to the present invention, will be explained in detail. Fig.1 shows a typical illustrative structure of a recording and/or reproducing apparatus 1 embodying the present invention. First, the portion of the structure responsible for recording externally input signals on a recording medium is explained. The recording and/or reproducing apparatus 1 is able to be fed with and record analog or digital data.

[0030] Analog video signals and analog audio signals are fed to terminals 11, 12, respectively. The video signals input to the terminal 11 are output to an analysis unit 14 and to an AV encoder 15. The audio signals input to the terminal 12 are output to the analysis unit 14 and to the AV encoder 15. The audio signals fed to the terminal 12 are output only to the AV encoder 15. The analysis unit 14 extracts feature points, such as scene changes, from the input video signals.

[0031] The AV encoder 15 encodes input video and

audio signal to output the system information (S), such as an encoded video stream (V), an encoded audio stream (A) and AV synchronization, to a multiplexer 16.

[0032] The encoded video stream is a video stream encoded e.g., with the MPEG (Moving Picture Expert Group) 2 system, whilst the encoded audio stream is an audio stream encoded in accordance with the MPEG1 system. The encoded audio stream is e.g., an audio stream encoded in e.g., the MPEG1 system or an audio stream encoded in accordance with the Dolby AC3 (trademark) system. The multiplexer 16 multiplexes the input video and audio streams, based on the input system information, to output a multiplexed stream through a switch 17 to a multiplexed stream analysis unit 18 and to a source packetizer 19.

[0033] The multiplexed stream is e.g., an MPEG-2 transport stream or an MPEG2 program stream. The source packetizer 19 encodes the input multiplexed stream into an AV stream composed of source packets in accordance with an application format of a recording medium 100 on which to record the stream. The AV stream is processed in ECC (error correction and coding) unit 20 and a modulation unit 21 with appendage of ECC codes and with modulation, before being output to a write unit 22, which then writes (records) an AV stream file based on a control signal output by the controller 23.

[0034] The transport stream, such as digital television broadcast, input from a digital interface or a digital television tuner, is input to a terminal 13. There are two recording systems for recording the transport stream, input to the terminal 13, one being a transparent recording system and the other being a system in which recording is preceded by re-encoding aimed at e.g., lowering the recording bit rate. The recording system designating information is input from a terminal 24 as a user interface to a controller 23.

[0035] In the transparent recording of the input transport stream, a transport stream, input to a terminal 13, is output to a multiplexed stream analysis unit 18 and to the source packetizer 19. The ensuing processing of recording an AV stream on a recording medium is the same as that of encoding and recording input audio and video signals, as described above, and hence is not explained here for simplicity.

[0036] If the input transport stream is re-encoded and subsequently recorded, the transport stream, input to the terminal 13, is input to a demultiplexer 26. The demultiplexer 26 demultiplexes the input transport stream to extract the video stream (V), the audio stream (A) and the system information (S).

[0037] Of the stream (information) extracted by the demultiplexer 26, the video stream is output to the AV decoder 27, while the audio stream and the system information are output to the multiplexer 16. The AV decoder 27 decodes the input video stream to output the reproduced video stream to the AV encoder 15, which then output the encoded video stream V to the multiplexer 16.

[0038] On the other hand, the audio stream and the system information, output from the demultiplexer 26 and input to the multiplexer 16, are multiplexed, based on the input system information, and output as a multiplexed stream to the multiplexed stream analysis unit 18 and to the source packetizer 19 through switch 17. The ensuing processing until recording of the AV stream on the recording medium 100 is similar to that of encoding and recording the video signals and hence is not explained specifically.

[0039] The recording and/or reproducing apparatus 1 of the present embodiment records a file of the AV stream on the recording medium 100, while also recording the application database information which accounts for the file. The application database information is prepared by the controller 23. The input information to the controller 23 is the feature information for the moving picture from the analysis unit 14, the feature information of the AV stream from the multiplexed stream analysis unit 18 and the user command information input at a terminal 24.

[0040] The feature information of the moving picture, supplied from the analysis unit 14, is the information pertinent to a picture characteristic of the input moving picture signals, and may, for example, include the specifying information (marks), such as a program start point, a scene change point or start and end points of the commercial (CM), or the thumbnail picture information of a picture of the specified site.

[0041] The feature information of the AV stream from the multiplexed stream analysis unit 18 is the information pertinent to the encoding information of the AV stream for recording, such as the address information of an I-picture in an AV stream, encoding parameters of the AV stream, transition point information of the encoding parameters in the AV stream, or the information pertinent to characteristic pictures in a video stream (marks).

[0042] The specifying information of the user from terminal 24 is the information in the AV stream, such as the specifying information of the playback domain as specified by the user, character letters which account for the contents of the playback domain or the bookmarks or resuming points for the user to set his or her favorite scene.

[0043] Based on the aforementioned input information, the controller 23 creates a database of the AV stream (Clip), a database of a group (Playlist) of playback domains (PlayItem) of the AV stream, management information of the recorded contents of the recording medium 100 (info.dvr) and the information on thumbnail pictures. Similarly to the AV stream, the application database information, constructed from the above information, is processed in the ECC unit 20 and the modulation unit 21 and input to the write unit 22, which then records a database file on the recording medium 100.

[0044] The above-described application database information will be explained subsequently in detail.

[0045] When the AV stream file recorded on the recording medium 100 (files of picture data and speech data) and the application database information, thus recorded on the recording medium 100, are reproduced, the controller 23 first commands a readout unit 28 to read out the application database information from the recording medium 100. The readout unit 28 reads out the application database information from the recording medium 100, which then reads out the application database information from the recording medium 100 to send the application database information through demodulation and error correction processing by a demodulating unit 29 and an ECC decoder 30 to the controller 23.

[0046] Based on the application database information, the controller 23 outputs a list of PlayList recorded on the recording medium 100 to a user interface of the terminal 24. The user selects the PlayList, desired to be reproduced, from the list of PlayLists. The information pertinent to PlayList, specified to be reproduced, is input to the controller 23. The controller 23 commands the readout unit 28 to read out the AV stream file necessary in reproducing the PlayList. In accordance with the command, the readout unit 28 reads out the corresponding AV stream from the recording medium 100 to output the read-out AV stream to the demodulating unit 29. The AV stream, thus input to the demodulating unit 29, is demodulated by preset processing and output through the processing by the ECC decoder 30 to the source packetizer 19.

[0047] The source depacketizer 31 converts the AV stream of the application format, read out from the recording medium 100 and processed in a preset fashion, into a stream that can be output to the demultiplexer 26. The demultiplexer 26 outputs the system information (S), such as the video stream (V), audio stream (A) or the AV synchronization, forming the playback domain (PlayItem) of the AV stream specified by the controller 23, to the AV decoder 27, which AV decoder 27 decodes the video stream and the audio stream to output the playback video signal and the playback audio signal to associated terminals 32, 33, respectively.

[0048] If fed from the terminal 24, as a user interface, with the information instructing random access playback or special playback, the controller 23 determines the readout position of the AV stream from the recording medium 100, based on the contents of the database (Clip) of the AV stream, to command the readout unit 28 to read out the AV stream. If the PlayList as selected by the user is to be reproduced as from a preset time point, the controller 23 commands the readout unit 28 to read out data from an I-picture having a time stamp closest to the specified time point.

[0049] If the user has commanded fast forward playback, the controller 23 commands the readout unit 28 to sequentially read out I-picture data in the AV stream in succession based on the database (Clip) of the AV stream.

[0050] The readout unit 28 reads out data of the AV stream from a specified random access point. The so read-out data is reproduced through processing by various components on the downstream side.

[0051] The case in which the user edits the AV stream recorded on the recording medium 100 is now explained. If desired to specify a playback domain for the AV stream recorded on the recording medium 100, for example, if desired to create a playback route of reproducing a portion sung by a singer A from a song program A, and subsequently reproducing a portion sung by the same singer A from another song program B, the information pertinent to a beginning point (IN-point) and an end point (OUT-point) of the playback domain is input to the controller 23 from the terminal as a user interface.

The controller 23 creates a database of the group (PlayList) of playback domains (PlayItem) of the AV streams. [0052] When the user desired to erase a portion of the AV stream recorded on the recording medium 100, the information pertinent to the IN-point and the OUT-point of the erasure domain is input to the controller 23, which controller 23 then modifies the database of the PlayList so as to refer to only the needed AV streams. The controller 23 also commands the write unit 22 to erase an unneeded stream portion of the AV stream.

[0053] The case in which the user desires to specify playback domains of an AV stream recorded on the recording medium 100 to create a new playback route, and to interconnect the respective playback domains in a seamless fashion, is now explained. In such case, the controller 23 creates a database of a group (PlayList) of the playback domains (PlayItem) of the AV stream and undertakes to partially re-encode and remultiplex the video stream in the vicinity of junction points of the playback domains.

[0054] The picture information at the IN-point and that at the OUT-point of a playback domain are input from a terminal 24 to a controller 23. The controller 23 commands the readout unit 28 to read out data needed to reproduce the pictures at the IN-point and at the OUT-point. The readout unit 28 reads out data from the recording medium 100. The data so read out is output through the demodulating unit 29, ECC decoder 30 and the source depacketizer 31 to the demultiplexer 26.

[0055] The controller 23 analyzes data input to the demultiplexer 26 to determine the re-encoding method for the video stream (change of picture\_coding\_type and assignment of the quantity of encoding bits for re-encoding) and the re-multiplexing system to send the system to the AV encoder 15 and to the multiplexer 16.

[0056] The demultiplexer 26 then separates the input stream into the video stream (V), audio stream (A) and the system information (S). The video stream may be classed into data input to the audio decoder 27 and data input to the multiplexer 16. The former is data needed for re-encoding, and is decoded by the AV decoder 27, with the decoded picture being then re-encoded by the AV encoder 15 and thereby caused to become a video

stream. The latter data is data copied from an original stream without re-encoding. The audio stream and the system information are directly input to the multiplexer 16.

[0057] The multiplexer 16 multiplexes an input stream, based on the information input from the controller 23, to output a multiplexed stream, which is processed by the ECC unit 20 and the modulation unit 21 so as to be sent to the write unit 22. The write unit 22 records an AV stream on the recording medium 100 based on the control signals supplied from the controller 23.

[0058] Data recorded on the recording medium 100 in such recording and/or reproducing apparatus 1 (data reproduced from the recording medium 100) is reproduced. Such data obtained on causing a bitstream, encoded in accordance with an encoding system, such as the MPEG Video or MPEG Audio, and multiplexed in accordance with the MPEG-2 system, to become a file handled by a file system, and on recording the file on a disc, is termed an AV (audio video) stream file or a Clip AV stream.

[0059] The scheme of specifying such AV stream file partially or entirely, arraying the specified portion for playback, by way of specifying the playback sequence, is hereinafter explained. The playback sequence delineation of specifying the AV stream file partially or entirely and of reproducing only the necessary portions is the PlayList. The PlayList is a self-contained unit in the user's perspective. The simplest structure is a unit from the start of recording until the end of recording. Lacking the editing, this unit becomes one PlayList.

[0060] The PlayList is comprised of designation of the AV stream file to be reproduced and a set of a playback beginning point (IN-point) and a playback end point (OUT-point). A set of the AV stream file, playback start point and the playback end point is termed the PlayItem. Thus, the PlayList is a set of PlayItems.

[0061] Referring to Fig.2, the PlayItem references a range specified by IN-point and OUT-point of a given AV stream. Reproducing a PlayItem is tantamount to reproduction of a portion of the AV stream referenced by the PlayItem.

[0062] While the AV stream is a bitstream multiplexed e.g., in the form of the MPEG-TS, the information having the relation of one-for-one correspondence (clip information) is held in a file distinct from the file in which the AV stream is recorded. This clip information is provided to facilitate reproduction and editing. This clip information and the AV stream is treated as one unit (object) and is termed a Clip. That is, the Clip is an object made up of the AV stream file and the ancillary information.

[0063] Fig.3 shows the above-described relation in the form of a UML diagram. The structure of the AV stream file, Clip, PlayItem and PlayList, represented by the UML diagram of Fig.3, renders possible the non-destructive playback sequence of reproducing only optional portions without modifying the AV stream file.

[0064] The file arrangement on a recording medium for recording or reproducing variable information for realization of the present invention is now explained. On the recording medium are recorded five sorts of files, namely info.dvr, menu.thmb (mark.thmb), #####.rpls (#####.vpls), where ##### denotes an optional number, %%%%.cpli, where %%%% denotes an optional number, and %%%%.clpi, where %%%% is such a number which causes each m2ts file to bear a one-for-one correspondence with the cpli file.

[0065] A directory /DVR is provided on a disc, so that a range underlying this directory is a range supervised by one disc respective system. It is noted that /DVR may be in a root directory of a disc or below a preset directory. In the /DVR directory are put info.dvr, menu.thmb and mark.thmb files. Below /DVR are put /PLAYLIST, /CLIPINF and /M2TS. The files #####.rpls and #####.vpls are put below /PLAYLIST, whilst %%%%.cpli and %%%%.m2ts are put in the /CLIPINF and /M2TS, respectively.

[0066] A sole file, that is a file info.dvr, is put below /DVR. The structure of info.dvr is represented by a syntax shown in Fig.5. The inside of a file is made up of blocks from one functional information to another, such that the information pertinent to the volume, the information pertinent to the arraying of the PlayLists and the information proper to the maker of the recording and/or reproducing apparatus 1 are stored in the DVRVolume(), TableOfPlayList() and in the MakersPrivateData(), respectively. In a leading end of a file are stated addresses indicating the leading ends of the respective blocks. For example, TableOfPlayList\_Start\_address indicates the position of beginning of TableOfPlayLists() in terms of the relative number of bytes in the file.

[0067] The structure of DVRVolume() is represented by the syntax shown in Fig.6. The version\_number and length represent the version number of DVRVolume() and the length in bytes from a field directly following the length to the trailing end of the DVRVolume(). ResumeVolume() is an area in which to store the information pertinent to resuming, whereas UIAppInfoVolume() is an area in which to store the attribute information of Volume.

[0068] Fig.7 shows the syntax of UIAppInfoVolume(). The character\_set indicates the method of encoding character letters encoded in the Volume\_name field. The encoding method may be selected from ASCII, Unicode and so forth. The name\_length denotes the length of bytes of the volume (disc) name indicated in the Volume\_name field. The Volume\_name denotes the name of the volume (disc). The number of bytes of the number of name\_length from left of the field is the effective character letter indicating the name of the volume (disc).

[0069] In the Volume\_name field, the values following these effective character letters are optional. The Volume\_protect\_flag is a flag indicating whether or not the contents in the volume may be presented to the user

for viewing without limitations. If 1 is set in this flag, the user may view the contents of the volume only if the user has succeeded in inputting the correct PIN number (PIN in Fig.7). If the flag is set to 0, the user may view the contents of the volume even if the user does not input the PIN number. If, when the user has inserted a disc into a player (recording and/or reproducing apparatus 1), the flag is set to 0, or the flag is set to 1 but the user has succeeded in correctly entering the PIN number, the player demonstrates a list in the PlayList in the so set disc.

**[0070]** The foregoing is directed to playback limitations imposed on the Volume. The playback limitations on the respective PlayLists are irrelevant to the Volume\_protect\_flag and are indicated by the playback\_control\_flag defined in the UIAppInfoPlayList () of Fig.9, as explained subsequently. PIN is four numerical figures of from 0 to 9, each of the four numerical figures being encoded in accordance with the ISO/IEC 646. If there exists a thumbnail representative of the Volume, ref\_thumbnail\_index is an area in which to store the number used for identifying the thumbnail. The thumbnail in the menu and thmb files having a thumbnail\_index specified by the ref\_thumbnail\_index becomes a menu thumbnail representing the Volume. In the present invention, thumbnails representing the Volume or PlayList are termed menu thumbnails.

**[0071]** The rp\_info\_valid\_flag, if 1, indicates that next following rp\_ref\_to\_Playlist\_file\_name, rp\_ref\_to\_Playitem\_id and rp\_time\_stamp are of valid values. The rp\_ref\_to\_PlayList\_file\_name indicates that the menu thumbnail representing the Volume has been created from a picture extracted from the pictures of a preset PlayList, and indicates the name of the PlayList file.

**[0072]** The rp\_ref\_to\_PlayItem\_id indicates PlayItem\_id indicating one PlayItem in the PlayList denoted by rp\_ref\_to\_PlayList\_file\_name, and indicates that the menu thumbnail representative of the Volume is created from the picture extracted from the picture of the PlayItem. The rp\_time\_stamp indicates the presentation time stamp of a picture in the PlayItem indicated by the rp\_ref\_to\_PlayTime\_id, and denotes that the menu thumbnail representative of the Volume has been created from the picture.

**[0073]** Fig.8 shows the syntax of TableOfPlayList(). It is noted that the number\_of\_PlayLists indicates the number of PlayLists in the Volume, with the PlayList\_file\_name specifying a filename, such as #####.rpls or #####vpls. In the UIAppInfoPlayList() are written various attributes of the PlayList, with the syntax being as shown in Fig.9. In the UIAppInfoPlayList(), shown in Fig. 9, there are stored a variety of attribute information not directly necessary for playback of PlayList, such as PlayList name, recording date and time, recording time or erasure inhibition or allowance. Of these, ref\_thumbnail\_index is able to specify a thumbnail as a representative picture of the PlayList. That is, the thumbnail in the file menu.thmb, having the thumbnail\_

index specified by ref\_thumbnail\_index is the thumbnail representative of the PlayList.

**[0074]** If rp\_info\_valid\_flag is 1, it indicates that the values of the next following rp\_ref\_to\_PlayTime\_id and rp\_time\_stamp are valid. On the other hand, rp\_ref\_to\_PlayItem\_id indicates the PlayItem\_id specifying one PlayItem in the PlayList and denotes that the menu thumbnail representative of the PlayList is created from the picture extracted from the picture in the PlayItem, whereas rp\_time\_stamp indicates the presentation time stamp of a picture in the PlayItem specified by the rp\_ref\_to\_Playitem\_id and denotes that the menu thumbnail representative of the PlayList is created from the picture.

**[0075]** One file %%%%.clpi shown in Fig.4 is created below /CLIPINF in association with each AV stream file %%%%.m2ts. The structure of %%%%.clpi is shown in Fig.10. The inside of the file is made up of blocks from one functional information to another. The information pertinent to Clip is stored in ClipInfo(), the information pertinent to discontinuous points of PCR (program clock reference) representing time base of the transport stream in the MPEG-2 system is stored in STC\_Info(), the information pertinent to the program in the MPEG-2 system is stored in ProgramInfo(), the information pertinent to CPI (Characteristic Point Information) representing feature points such as random access start point in the AV stream is stored in CPI() and the mark information such as index points or commercial start and end points affixed to the Clip is stored in Clip-Mark. In the leading end of the file are recorded addresses indicating the leading ends of the respective blocks.

**[0076]** Here, only the blocks relevant to thumbnails are explained. In the present embodiment, the operation of affixing the thumbnails to the clip is realized by affixing the thumbnails to the clip marks. The information of the marks affixed to the clips is stored in the ClipMark(). Fig. 11 shows the syntax of the ClipMark(). In the syntax, mark\_type denotes mark types, such as resume, bookmark or skip, while mark\_time\_stamp indicates the time of affixing the mark. The field pertinent to the thumbnail is ref\_thumbnail\_index. It is noted that ref\_thumbnail\_index specifies the thumbnail number to specify the thumbnail in the mark.thmb file in which the thumbnail data is stored. The thumbnail specified by ref\_thumbnail\_index becomes the mark thumbnail afforded to the mark. In the present invention, the thumbnail afforded to the mark is specifically termed a mark thumbnail.

**[0077]** The mark is mainly used to specify a highlight or a characteristic scene in the Clip and in the PlayList. The skip function of commanding skipping the playback as from the mark to reproduce the next PlayItem is also realized by the mark. Fig.12 shows that the mark is accorded to the Clip and to the PlayList. The features of the mark are now summarized.

**[0078]** The mark added to the Clip specifies characteristic scenes attributable to the contents of the AV



stream, for example, scene change points. In reproducing the PlayList, random access may be made using the mark in the Clip being referenced by the PlayList. In Fig. 12, two marks of commercial (CM) and scene are afforded to the Clip. The commercial mark is utilized from the 5 least PlayList, whilst the scene mark is utilized from two PlayLists. The marks added to the PlayList, such as bookmarks or resume points, are mainly set by the user. These are bookmark and resume points in Fig. 12.

[0079] The marks are set in the Clip or PlayList by adding the time stamp indicating the mark time point to the mark list. Mark deletion is tantamount to removing the time stamp of the mark from the mark list. Therefore, mark setting or deletion causes no changes in the AV stream.

[0080] The files #####.rpls and #####.vpls, shown in Fig. 4, are hereinafter explained. One of the files #####.rpls and #####.vpls is created under the /PLAYLIST for each PlayList. The structure of files #####.rpls and #####.vpls is shown in Fig. 13. The inside of the file is made up of blocks from one functional information to another. The information pertinent to PlayList is stored in PlayList(), the information of the mark afforded to the PlayList is stored in the PlayListMark() and the information proper to the maker of the recording and/or reproducing apparatus having recorded the PlayList file is stored in the MakersPrivateData(). In the leading end of the file are recorded addresses indicating the leading ends of blocks, such as PlayListMark\_Start\_address. So, padding\_byte may be inserted ahead or at back of the block. However, the beginning position of the PlayList() is fixed and is set to e.g., 256th byte as from the leading end of the file.

[0081] The contents of the block PlayList() is shown in Fig. 14. First and foremost is version\_number indicating the version number of the next following information. The length indicates the byte length as from the field directly following the length to the end of the PlayList(). The PlayList\_type indicates the sort of the PlayList and CPI\_type denotes the type of the CPI owned by this PlayList. The number\_of\_PlayItems denotes the number of PlayItems forming the PlayList, while the number\_of\_SubPlayItems denotes the number of Play-items (SubPlayItem) for post recording afforded to the PlayList. The PlayItem() and the SubPlayItem() store the information of the Playitem and the information of the SubPlayitem, respectively.

[0082] The contents of the block PlayItem() are as shown in Fig. 15. In the Clip\_Information\_file\_name, the filenames of the Clip information files (files having the extender dpi) referenced by the PlayItem are stored in the form of letter strings. STC\_sequence\_id indicates the domain of the time range with continuous PCR. Since an integrated continuous time axis may be defined in this domain, the start and end points of the PlayItem can be determined uniquely. That is, the start and end points of the respective PlayItems must be present in the same STC\_sequence.

[0083] IN\_time denotes pts (presentation time stamp) on the STC\_sequence of the start point of this PlayItem, whilst OUT\_time denotes the pts on the STC\_sequence of the end point of the PlayItem. The connection\_condition is the information indicating how this Playitem is coupled to the next Playitem and denotes the condition as to whether or not the neighboring Playitems can be reproduced seamlessly.

[0084] BridgeSequenceInfo() stores the information pertinent to a bitstream (bridge sequence) generated when skipping to a bitstream different from the bitstream to be inherently reproduced at a junction of the Play-items and reproducing this different bitstream to realize the function of reproducing the neighboring Playitems seamlessly. The program\_number denotes a set of elementary streams, such as video or audio streams, defined in MPEG system, that is a channel in the television broadcast.

[0085] The above is the gist of the data structure forming the PlayList and the Playitem. With such data structure, it is possible to form the PlayList by an array of Playitems in the AV stream which are desired to be reproduced and which are specified by sets of IN and OUT points, while it is also possible to supervise a set of playback unit recognized by the user.

[0086] In the embodiment of the present invention, the operation of affixing a thumbnail at an optional time point on the PlayList is realized by affixing a thumbnail to the PlayList mark. The mark information affixed to the PlayList is stored in the PlayListMark(). Fig. 16 shows the syntax of the PlayListMark(). The mark\_type in the syntax denotes the type of the mark, such as resume, bookmark and skip, while mark\_time\_stamp denotes the time the mark is assigned. The ref\_thumbnail\_index specifies the thumbnail number to specify the thumbnail in the mark.thmb file in which thumbnail data is stored. The thumbnail specified by ref\_thumbnail\_index becomes the mark thumbnail corresponding to the mark.

[0087] The thumbnail is explained in detail. The thumbnail denotes a still picture ancillary to Volumes, PlayList and Clip. There are two sorts of the thumbnail, one being a thumbnail representing the contents. This thumbnail is mainly used in a menu picture in order to permit the user to select the picture desired to be viewed using a cursor. The other is a picture representing a scene pointed by the mark.

[0088] The Volume and each PlayList need to be able to have respective representative pictures. The representative picture of the Volume is presupposed to be used for such a case wherein a still picture indicating the disc contents are displayed first. The representative picture of the PlayList is presupposed to be used as a still picture for demonstrating the PlayList contents in the menu picture used in selecting the PlayList.

[0089] The simplest method of realizing the representative picture of the PlayList is to render the first picture of the PlayList a thumbnail. However, the leading picture at the playback time 0 is not necessarily optimum

in representing the contents. So, an optional picture is adapted for being determined as a thumbnail on the PlayList. The above two thumbnails are termed menu thumbnails. The menu thumbnails are demonstrated frequently and hence need to be able to be read out speedily from the disc. For satisfying the need, all menu thumbnails are stored in one file for the sake of efficiency. The menu thumbnails are not necessarily the pictures extracted from the moving pictures in the volume, but may be pictures captured from the personal computer or from a digital still camera.

[0090] On the other hand, it is necessary that plural marks can be entered in the Clip and PlayList and that, for grasping the contents of the mark positions, the mark point pictures need to be able to be viewed easily. The pictures indicating these mark points are termed mark thumbnails. So, the thumbnails are derived mainly from the extracted markpoint pictures, rather than the pictures captured from outside, as shown in Figs.18 and 19. In distinction from the menu thumbnails, the mark thumbnails are used e.g., in a submenu used for representing details of the PlayList, and hence need not be read out in short access time.

[0091] So, there is no problem if the operation of the player opening a file to read out a portion of the file each time a thumbnail is necessary is somewhat time-consuming. For decreasing the number of files present in the volume, all mark thumbnails are desirably stored in one file. Although the Play List may have one menu thumbnail and plural mark thumbnails, the Clip need not be selected directly by the user (it is usually specified through PlayList) so that it does not have the menu thumbnail. Fig.20 shows the above-described relation.

[0092] Since the thumbnails are added and deleted every so often, the operations of addition and partial deletion need to be executed readily and speedily. For this reason, the Thumbnail() has a block structure. Picture data are divided into plural portions each of which is stored in one tn\_block(). One picture data is stored in continuous tn\_block(). There may exist tn\_block() not in used in the string of the tn\_block(). The byte length of one thumbnail picture is variable.

[0093] Fig.21 shows a syntax of the file which stores thumbnail data. In this file, there exists only one Thumbnail(). Fig.22 shows the syntax of thumbnail(). The version\_number denotes four character letters indicating the version number of this thumbnail(). The length is a 32-bit unsigned integer indicating the number of bytes of thumbnail(), whilst tu\_blocks\_start\_address is a 32-bit unsigned integer indicating the start byte address as from the leading end of thumbnail().

[0094] The number\_of\_thumbnails is a 16-bit unsigned integer indicating the number of thumbnail pictures stored in this thumbnail (). The tu\_blocks\_start\_address is a 16-bit unsigned integer representing the size of 1 tn\_block() in kilobytes. For example, the tu\_blocks\_start\_address being 1 indicates that the size of one tn\_block() is 1024 bytes. The number\_of\_

tu\_blocks is a 16-bit unsigned integer indicating the number of the tn\_block() in this thumbnail().

[0095] The thumbnail\_index is a 16-bit unsigned integer indicating an index number of a thumbnail represented by the thumbnail information for one for-loop beginning from this thumbnail\_index field. The value of 0xFFF must not be used for thumbnail\_index. The thumbnail\_index is referenced by ref\_thumbnail\_index. The thumbnail\_picture\_format is an 8-bit unsigned integer indicating a picture format of a thumbnail picture and assumes a value shown in Fig.23. The DCF (Design Rule for camera File System) and the PNG (Portable Network Graphics) are allowed only within "menu.thmb". That is, the mark thumbnail must assume a value "0x00" (MPEG-2 Video I-picture).

[0096] The picture\_data\_size is a 32-bit unsigned integer indicating the byte length of an encoded thumbnail picture, while the start\_tn\_block\_number is a 16-bit unsigned integer indicating the tn\_block\_number of tn\_blocks with which begins data of a thumbnail picture. The leading end of thumbnail picture data must coincide with the leading end of tb\_block. The start\_tn\_block\_number begins with 0 and is relevant to the number of a variable k in the tn\_block for-loop. The x\_picture\_length is a 16-bit unsigned integer indicating the number of pixels in the horizontal direction of the thumbnail picture.

[0097] The y\_picture\_length is a 16-bit unsigned integer indicating the number of pixels in the vertical direction of the thumbnail picture. The tu\_block is an area in which to store picture data. All of tn\_blocks() in the thumbnail() must be of the same size prescribed in tn\_block\_size. Figs.24A and 24B schematically indicate the manner in which the picture data are stored in the tn\_block. As shown in Figs.24A and 24B, each picture data begins at the leading end of tn\_block(). If a picture data is of a size exceeding one tn\_block(), it is stored by exploiting the next following tn\_block().

[0098] Although picture data is of variable length, one tn\_block() is of a fixed size. By converting the variable length data into fixed length data, addition of new picture data or deletion of picture data can be dealt with on the block basis, thus simplifying address management.

[0099] An alternative proposal of the directory file structure and the syntax for thumbnail recording is hereinafter explained. An alternative proposal of the directory file structure shown in Fig.4 is shown in Fig.25. In Fig. 4, there is only one file for recording a mark thumbnail, namely mark.thmb, below /DVR. In Fig.25, plural files xxxxx.thmb are provided in a one-for-one correspondence for each PlayList and Clip (xxxxx is a filename, excluding an extender, representing an associated PlayList or Clip).

[0100] For example, the mark thumbnail picture, owned by the PlayList represented by #####.rpls, is stored in #####.thmb in the same directory. Similarly, the picture of the mark thumbnail owned by the Clip represented by %%%%.clpi, is stored in %%%%.thmb

in the same directory. The mark thumbnail of the PlayList clip is subjected to the constraint that it is stored in the .thmb file having the same name portion except the file extender, with the syntax in the xxxxx.thmb being the same as menu.thmb or mark.thmb (Fig.21), as shown in Fig.26.

**[0101]** As for the menu thumbnail of the PlayList, the ref\_thumbnail\_index specifying a thumbnail is placed in Figs.8 and 9 in UIAppInfoPlayList() in TableOfPlayLists() of an Info.dvr file. Alternatively, the UIAppInfoPlayList() may be placed not in Info.dvr but in each PlayList file. In such case, the UIAppInfoPlayList() is moved from TableOfPlayLists() to PlayList() in the PlayList file. So, the syntax shown in Fig.8 is as the syntax shown in Fig.27, whilst the syntax shown in Fig.14 is as the syntax shown in Fig.28. The UIAppInfoPlayList() in Fig.28 is as the syntax shown in Fig.29, in which is inserted ref\_thumbnail\_index.

**[0102]** Thumbnail recording is possible with the above-described file and the syntax.

**[0103]** Referring to the flowchart of Fig.30, creation of a mark thumbnail is explained. At step S1, the user selects the PlayList desired to be reproduced. At step S2, an AV stream starts to be reproduced, based on the selected PlayList. At step S3, the user views the AV stream being reproduced to search a scene desired to be marked. If the user has succeeded in searching the scene to be marked, the user acts on a mark button of a remote controller, not shown, annexed to the recording and/or reproducing apparatus 1. This operation may be made during playback or during pause.

**[0104]** If a mark button is acted on by the user at step S3, the mark position is determined at step S4 as a processing corresponding to user command. If the mark position is set, a picture which is to be a thumbnail picture is selected at step S5. At a time point when the mark position is specified, the controller 23 (Fig.1) may inquire of the user whether or not the thumbnail is to be formed and, if the thumbnail is to be formed, the controller 23 may furnish a candidate thumbnail picture to the user by e.g., intermittent feed to permit the user to select the thumbnail picture.

**[0105]** If the picture which is to be the thumbnail picture has been selected by the user or by the controller 23, the controller 23 captures a picture from the AV encoder 15 to transfer the picture to a RAM (random access memory), not shown, of the controller 23. At step S7, it is verified whether or not the picture is to be compressed. If the picture is verified to be compressed, the controller 23 transfers to step S8 to effect picture compression to return data to the RAM. If it is verified at step S7 that the picture is not to be compressed, the controller 23 skips the processing of step S8 to transfer to step S9.

**[0106]** At step S9, the controller 23 creates the header information, such as thumbnail encoding system, thumbnail\_id for identifying a thumbnail, size (number bytes) and the thumbnail header constituted by the

number of pixels in the X- and Y-directions, to divide the picture data into tu\_block() units. At step S10, picture data split into the header information and the tn\_block() is transferred to the ECC unit 20 through write unit 22 so as to be written as mark.thmb file on the recording medium 100.

**[0107]** Referring to the flowchart of Fig.31, the method for creating a menu thumbnail is explained. Since the menu thumbnail is assigned to the Volume or to each PlayList, the controller 23 at step S21 allows the user to select the object to which to afford the menu thumbnail. At step S22, the controller 23 allows the user to select whether a thumbnail is to be captured from outside or from a scene in the PlayList.

**[0108]** If it is verified at step S22 that the thumbnail is to be captured from outside, the controller 23 transfers to step S23 where picture data is captured from an input terminal specified by the user or from a file. When the capturing is finished, the controller 23 proceeds to step S26.

**[0109]** If it is verified at step S22 that the thumbnail is not to be captured from outside, the controller 23 transfers to step S24 where the controller 23 first allows the user to select from which PlayList the thumbnail picture is to be captured. The controller 23 then proceeds to reproduce the selected PlayList. At step S25, the controller 23 allows the user to select the scene desired to be rendered a thumbnail.

**[0110]** If the thumbnail picture has been selected at step S23 or step S25, the controller 23 proceeds to step S26. The processing as from step S26 is similar to the processing as from step S5 of the flowchart of Fig.30 and hence is not explained specifically.

**[0111]** By affording the thumbnail picture to the PlayList as described above, these thumbnail pictures are easier to supervise, thus facilitating, the processing of selecting the AV stream in reproducing the recorded AV stream.

**[0112]** Fig.32 shows another embodiment of the directory and the file to be created on the disc. The "menu.tidx" and "menu.tdat" store the information of a representative picture for each menu thumbnail, that is each picture and each PlayList representing the Volume. The header information of the totality of the menu thumbnails is managed as a compact menu.tidx. The picture data of the totality of the menu thumbnails is managed as a compact menu.tdat.

**[0113]** The "mark.tidx" and "mark.tdat" store the information pertinent to a mark thumbnail, that is a picture specified by a mark point. The header information of the totality of the mark thumbnails is managed in a compact mark.tidx, while picture data of the totality of the mark thumbnails is managed in a compact mark.tdat.

**[0114]** That is, in the present file structure, the thumbnail header information and the thumbnail picture data are managed in separate files. The reason is relevant to the file management method undertaken by the file system. That is, since the file of the thumbnail header

information is the relatively crucial information, the file system manages the data by dual writing on the disc.

[0115] The reason of doing this is to deal successfully with the case of one of the data being lost due to scratches on the disc, and has the meaning of data backup. The picture data file is not written in duplex because it is of relatively large volume such that the capacity required on the disc is increased due to dual writing.

[0116] The encoding method, sampling structure and scanning structure for picture data stored in the "menu.tdat" and "mark.tdat" can be changed from picture to picture. However, the recording and/or reproducing apparatus 1 can be simpler in structure if the totality of the picture data are the same. For example, the JFIF (JPEG file interchange format), component signals and progressive scan format are used.

[0117] The syntax and the semantics of these four files are explained. The "menu.tidx" and "mark.tdat" are of the same syntax structure. Fig.33 shows the syntax structure of "menu.tidx" and "mark.tdat". The version number is the four numerical figures indicating the version number of the thumbnail header information file.

[0118] The length is the number of bytes as a from the byte directly following the length field to the last byte of the menu.tidx/mark.tidx. In case of menu.tidx, the number\_of\_thumbnails is the number of thumbnails stored in the menu.tdat and, in case of the mark.tidx, it is the number of the thumbnail pictures stored in the mark.tdat.

[0119] In the cases of menu.tidx and mark.tidx, tn\_block\_size indicates the size of one tn\_block in menu.tdat and the size of one tn\_block in mark.tdat, respectively. This size is based on 1024 bytes as a unit. For example, tn\_block\_size = 1 indicates that the size of one tn\_block is 1024 bytes. One thumbnail picture must be stored in one tn\_block.

[0120] In the cases of menu.tidx and mark.tidx, the number\_of\_tn\_blocks indicates the number of tn\_blocks in the menu.tdat and that in the mark.tdat, respectively. The thumbnail\_index indicates the index number of the thumbnail information next following the thumbnail\_index field. As the thumbnail\_index, the value of 0xFFFF must be used.

[0121] In the case of menu.tidx, the thumbnail\_index is referenced by ref\_thumbnail\_index in UIAppInfoVolume and in UIAppInfoPlaylist. In the case of mark.tidx, the thumbnail\_index is referenced by ref\_thumbnail\_index in PlaylistMark() and in ClipMark().

[0122] In the case of menu.tidx, ref\_to\_block\_id indicates one tn\_block in menu.tdat, with the tn\_block storing picture data indicated by thumbnail\_index. The value of ref\_to\_block\_id references the value of tn\_block\_id in the menu.tdat syntax.

[0123] In the case of mark.tidx, ref\_to\_block\_id indicates one tn\_block in mark.tdat, with the tn\_block storing picture data indicated by thumbnail\_index. The value of ref\_to\_block\_id references the value of tn\_block\_id in the menu.tdat syntax.

[0124] The picture\_byte\_size indicates the data length of one encoded thumbnail picture pointed by thumbnail\_index in bytes. The picture\_byte\_size must be less than the value of  $1024 \times \text{tn\_block\_size}$ . That is, the encoding by the recording and/or reproducing apparatus 1 must be such that the data length of one encoded thumbnail picture will be not larger than  $1024 \pm \text{value of tn\_block\_size}$ .

[0125] The horizontal\_picture\_size indicates the number of pixels in the horizontal direction of the encoded thumbnail picture pointed by thumbnail\_index. The vertical\_picture\_size indicates the number of pixels in the vertical direction of the encoded thumbnail picture pointed by thumbnail\_index. The display\_aspect\_ratio shows a display aspect ratio of the encoded thumbnail picture pointed by thumbnail\_index. The meaning of the values is shown in Fig.34.

[0126] The color\_space indicates the format when converting component signals of Y, Cb and Cr into component signals of R, G and B. The meaning of the values is shown in Fig.35.

[0127] The "menu.tdat" and the "mark.tdat" are of the same syntax structure. Fig.36 shows the syntax structure of "menu.tdat" and the "mark.tdat". The tn\_block is an area in which to store one encoded thumbnail picture. The byte length of one thumbnail picture is not larger than the size of one tn\_block. The first byte of one picture data must coincide with the first byte of tn\_block.

[0128] In the cases of "menu.tdat" and "mark.tdat", the size of one tn\_block is indicated by the tn\_block\_size in menu.tdat and by the tn\_block\_size in mark.tdat, respectively. Each tn\_block is differentiated by the value of tn\_block\_id when the tn\_block appears in the syntax for-loop. In the cases of "menu.tidx" and "mark.tidx", tn\_block\_id is referenced by ref\_to\_tn\_block\_id in the menu.tidx and by ref\_to\_tn\_block\_id in the mark.tidx, respectively.

[0129] Since the thumbnails are added and deleted every so often, the operations of addition and partial deletion need to be executed readily and speedily. For this reason, "menu.tdat" and "mark.tdat" are of a block structure. One picture data is stored in one tn\_block().

[0130] A tn\_block not in use may be present in the tn\_block string of "menu.tdat" and "mark.tdat". For example, if, in deleting a preset thumbnail, the thumbnail\_index entered in a header information file of the thumbnail is erased, without changing the picture data file of the thumbnail, tn\_blocks not in use are produced in the tn\_block string.

[0131] Fig.37 schematically shows the manner in which thumbnail picture data are stored in the tn\_block. As shown in Fig.37, the byte length of one thumbnail picture is not larger than the size of one tn\_block. A tn\_block not in use may be present on one tn\_block.

[0132] In the above embodiment, it is assumed that menu.tbat and mark.tbat files are recorded as they are divided into two. Alternatively, each file may be conceived as being one block and a first block of menu.tbat

data and a second block of mark.tbat data may be recorded as being one compact file. In recording in one compact file, SequenceInfo, CPI and Clipmark may be recorded as separate blocks.

[0133] The above-described sequence of operations may be executed not only by hardware but also by software. If the sequence of operations is to be executed by software, it is installed from a recording medium to a computer in the dedicated hardware of which the program forming the software is installed or to a general-purpose personal computer of Fig.38 capable of executing various functions based on a variety of programs installed therein.

[0134] In the personal computer, shown in Fig.38, a CPU (central processing unit) 201 executes a variety of operations in accordance with programs stored in a ROM (read-only memory) 202 or programs loaded from a memory 208 to a RAM (random-access memory) 203. There is also stored in the RAM 203 data required in executing various processing operations.

[0135] The CPU 201, ROM 202 and the RAM 203 are interconnected over a bus 204. An input/output interface 205 is also coupled to the bus 204.

[0136] To the input/output interface 205 are coupled an input unit 206, comprised of a keyboard and a mouse, a display comprised of CRT and an LCD, an output unit 207, comprised of a speaker, a memory 208, comprised of a hard disc and a communication unit 209 comprised of a modem and a terminal adapter. The communication unit 209 executes communication processing over a network.

[0137] To the input/output interface 205, there are also connected a drive 210, a magnetic disc 221, an optical disc 222, a magneto-optical disc 223 or a semiconductor memory 224. The program read out therefrom is installed as necessary on the memory 208.

[0138] The recording medium is constituted not only by a package medium distributed for furnishing the program to the user, in addition to a computer, such as a magnetic disc 221 carrying the program thereon, inclusive of a floppy disc, an optical disc 222 inclusive of a CD-ROM (Compact Disc-Read-Only memory) or a DVD (Digital Versatile Disc), a magneto-optical disc, inclusive of a Mini-Disc, or a semiconductor memory 224, but also by a hard disc, inclusive of a ROM 202 carrying a program and a memory 208, furnished to the user as it is built-in in a computer, as shown in Fig.38.

[0139] In the present specification, the steps of the program furnished by the medium include not only the chronological processing in accordance with the sequence indicated, but also the processing not chronologically but in parallel or separately.

[0140] Additionally, in the specification, the system means an entire apparatus comprised of plural component devices.

## Industrial Applicability

[0141] In the recording method and apparatus and in the first program, according to the present invention, a thumbnail picture representing the moving picture data is generated from input moving picture data as first thumbnail data. From the moving picture data, a thumbnail picture of a picture extracted as being a characteristic picture, or a thumbnail picture of a picture specified by a user, is generated as second thumbnail data, and the first and second thumbnail data are recorded as respective independent groups on the recording medium. So, the user is allowed to select recorded data readily using the recorded picture data.

[0142] In the reproducing method and apparatus and in the second program, according to the present invention, if playback of an AV stream is commanded, management data supervising picture data of a thumbnail picture representing the contents of the AV stream is read out, and the picture data is read out based on the management data read out by readout means. From the AV stream, picture data of a thumbnail picture of a picture extracted from it as a characteristic, picture or picture data of a thumbnail picture of a picture specified by a user, and management data supervising the picture data, are read out, and the picture data are read out based on the read-out management data. So, the user is allowed to select recorded data readily using the read-out management data.

## Claims

### 1. A recording apparatus comprising:

means for generating, from input moving picture data, a thumbnail picture representing said moving picture data, as first thumbnail data, and for generating, from said moving picture data, a thumbnail picture of a picture extracted as a characteristic picture, or a thumbnail picture of a picture specified by a user, as a second thumbnail data; and  
means for recording said first thumbnail data and the second thumbnail data, generated by said generating means as independent groups on a recording medium.

### 2. The recording apparatus according to claim 1 wherein said generating means generates said first and second thumbnail data as respective independent files.

### 3. The recording apparatus according to claim 1 wherein said generating means generates said first and second thumbnail data as respective independent data blocks and as one file.

4. The recording apparatus according to claim 1  
wherein said generating means generates first  
management data supervising said first thumbnail  
data with a number corresponding to the first  
thumbnail data, said generating means also gener- 5  
ating second management data supervising said  
second thumbnail data with a number correspond-  
ing to the second thumbnail data;  
said recording means recording said first and  
second management data in said recording medi- 10  
um.
5. The recording apparatus according to claim 4  
wherein said first and second management data in- 15  
clude data specifying the format of the picture data  
of the thumbnail data being supervised.
6. The recording apparatus according to claim 4  
wherein said recording means records picture data  
of said thumbnail picture contained in said first or 20  
second management data in terms of a block of a  
preset size as a unit.
7. The recording apparatus according to claim 1  
wherein said recording means records the informa- 25  
tion representing the referencing destination of said  
first thumbnail data as a separate file on said re-  
cording medium.
8. The recording apparatus according to claim 1 30  
wherein said recording means further records the  
information indicating the referencing destination of  
said thumbnail picture contained in said second  
thumbnail data. 35
9. A recording method comprising:  
a generating step of generating, from input  
moving picture data, a thumbnail picture repre- 40  
senting said moving picture data, as first  
thumbnail data, and for generating, from said  
moving picture data, a thumbnail picture of a  
picture extracted as a characteristic picture, or  
a thumbnail picture of a picture specified by a 45  
user, as a second thumbnail data; and  
a recording controlling step of exercising con-  
trol for recording said first thumbnail data and  
the second thumbnail data, generated by said  
generating step as independent groups on a re- 50  
cording medium.
10. A recording medium having recorded thereon a  
computer-readable program, said computer-reada-  
ble program comprising: 55  
a generating step of generating, from input  
moving picture data, a thumbnail picture repre-  
senting said moving picture data, as first  
thumbnail data, and for generating, from said  
moving picture data, a thumbnail picture of a  
picture extracted as a characteristic picture, or  
a thumbnail picture of a picture specified by a  
user, as a second thumbnail data; and  
a recording controlling step of exercising con-  
trol for recording said first thumbnail data and  
the second thumbnail data, generated by said  
generating step as independent groups on a re-  
cording medium.
11. A program for having a computer execute  
a generating step of generating, from input  
moving picture data, a thumbnail picture represent-  
ing said moving picture data, as first thumbnail data,  
and for generating, from said moving picture data,  
a thumbnail picture of a picture extracted as a char-  
acteristic picture, or a thumbnail picture of a picture  
specified by a user, as a second thumbnail data; and  
a recording controlling step of recording said  
first thumbnail data and the second thumbnail data,  
generated by said generating step as independent  
groups on a recording medium.
12. A reproducing apparatus comprising:  
first readout means for reading out manage-  
ment data supervising picture data of a thumb-  
nail picture representing the contents of said  
picture data in case playback of said picture da-  
ta is commanded;  
second readout means for reading out said pic-  
ture data based on said management data read  
out by said readout means;  
third readout means for reading out picture data  
of a thumbnail picture of a picture extracted as  
a characteristic picture from said picture data  
or picture data of a thumbnail picture of a pic-  
ture specified by a user, and management data  
supervising said picture data; and  
fourth readout means for reading out said pic-  
ture data based on said management data read  
out from said third readout means.
13. A reproducing method comprising:  
a first readout controlling step of reading out  
management data supervising picture data of  
a thumbnail picture representing the contents  
of said picture data in case playback of said pic-  
ture data is commanded;  
a second readout controlling step of controlling  
the readout of said picture data based on said  
management data readout-controlled by the  
processing of said first readout control means;  
a third readout controlling step of reading out  
picture data of a thumbnail picture of a picture

extracted as a characteristic picture from said picture data or picture data of a thumbnail picture of a picture specified by a user, and management data supervising said picture data; and  
 a fourth readout controlling step of reading out said picture data based on said management data readout-controlled by the processing of said third readout control step.

5

10

14. A recording medium having recorded thereon a computer-readable program, said computer-readable program comprising:

a first readout controlling step of controlling the readout of management data supervising picture data of a thumbnail picture representing the contents of said picture data in case playback of said picture data is commanded;  
 a second readout controlling step of controlling the readout of said picture data based on said management data readout-controlled by the processing of said first readout control means;  
 a third readout controlling step of reading out picture data of a thumbnail picture of a picture extracted as a characteristic picture from said picture data or picture data of a thumbnail picture, of a picture specified by a user, and management data supervising said picture data; and  
 a fourth readout controlling step of reading out said picture data based on said management data readout-controlled by the processing of said third readout controlling step.

15

20

25

30

35

15. A program for having a computer execute

a first readout controlling step of controlling the readout of management data supervising picture data of a thumbnail picture representing the contents of said picture data in case playback of said picture data is commanded;

40

a second readout controlling step of controlling the readout of said picture data based on said management data readout-controlled by the processing of said first readout control means;

45

a third readout controlling step of reading out picture data of a thumbnail picture of a picture extracted as a characteristic picture from said picture data or picture data of a thumbnail picture of a picture specified by a user, and management data supervising said picture data; and

50

a fourth readout controlling step of reading out said picture data based on said management data readout-controlled by the processing of said third readout control step.

55

16. A recording medium having recorded thereon first data comprised of picture data, thumbnail picture

data of a thumbnail picture extracted from said picture data, indicating the contents of the picture data, and management data for supervising data of said thumbnail picture, and second data comprised of thumbnail picture data of a thumbnail picture of a picture extracted as a characteristic picture from said first stated picture data or picture data of a thumbnail picture of a picture specified by the user and management data supervising the picture data.

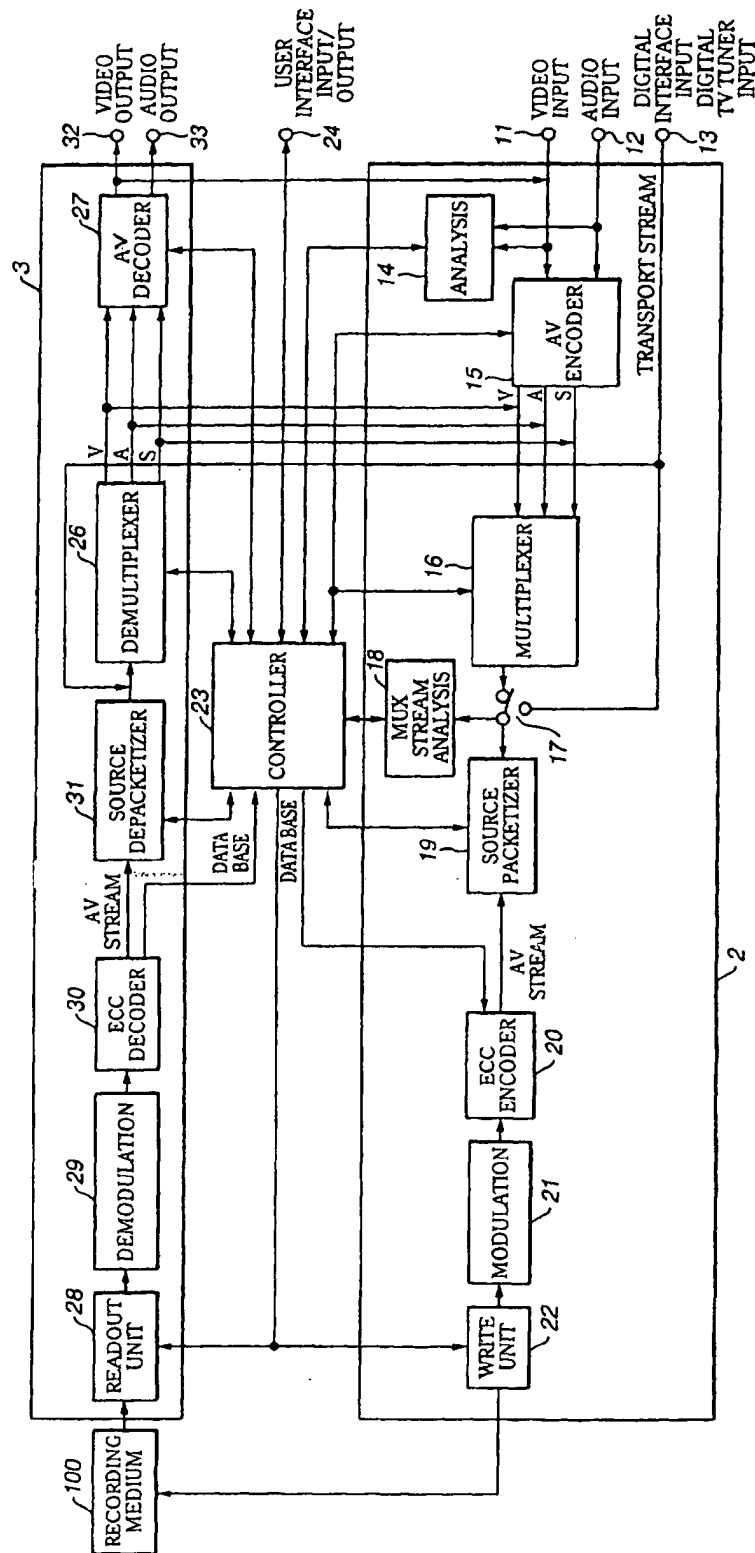


FIG.1



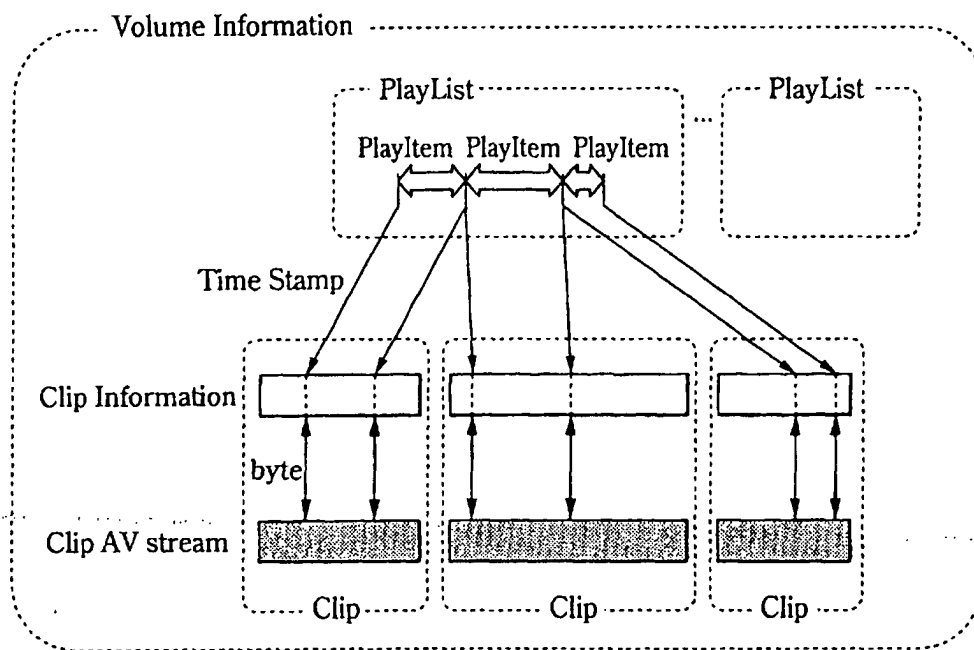
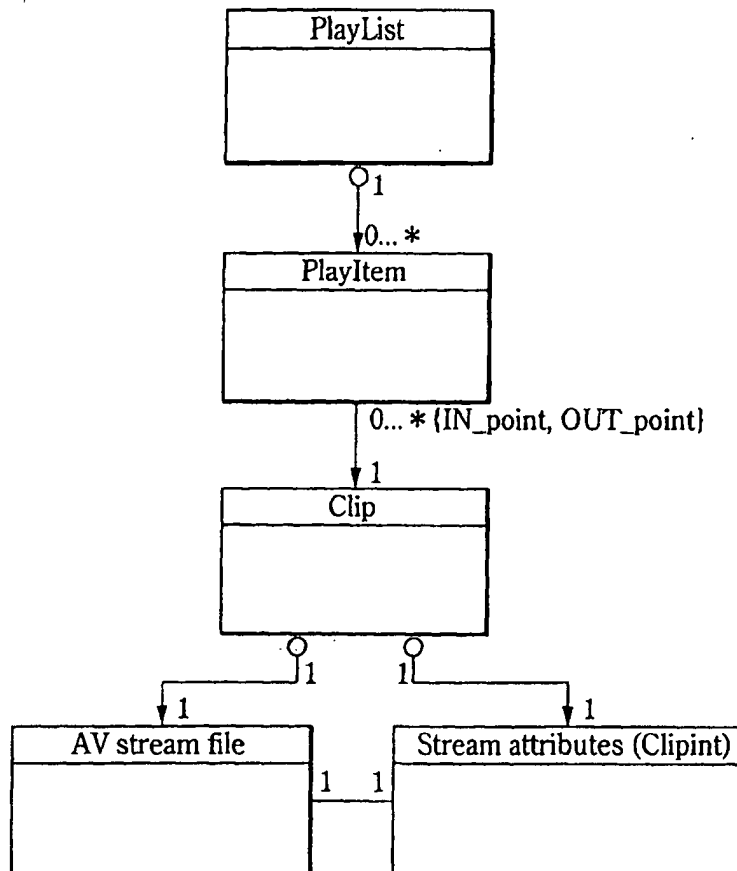
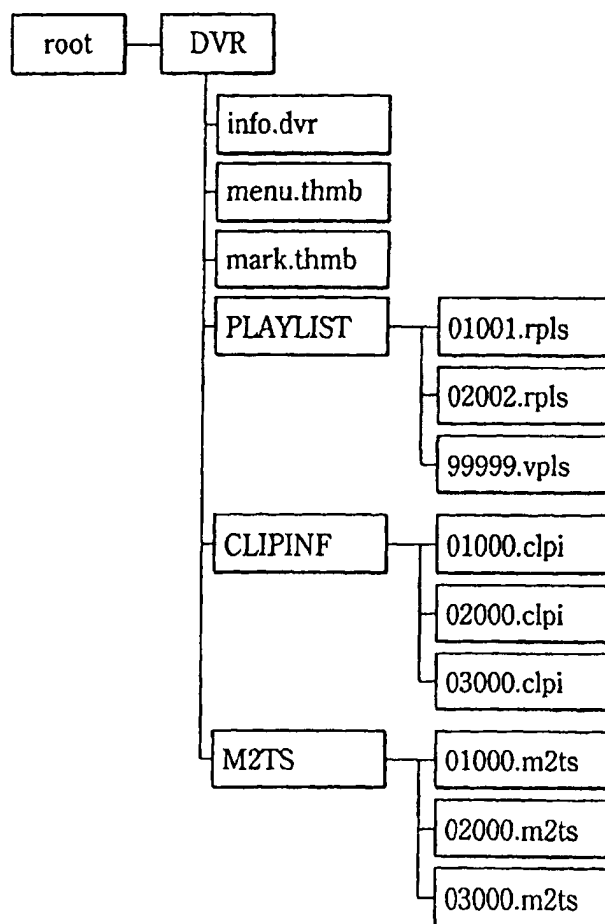


FIG.2



**FIG.3**



**FIG.4**

| SYNTAX                                | NUMBER<br>OF BYTES | ABBREVIATION |
|---------------------------------------|--------------------|--------------|
| info.dvr {                            |                    |              |
| <b>TableOfPlayLists_Start_address</b> | 32                 | uimsbf       |
| <b>MakerPrivateData_Start_address</b> | 32                 | uimsbf       |
| reserved                              | 192                | bslbf        |
| <b>DVRVolume()</b>                    |                    |              |
| for (i=0;i<N1;i++){                   |                    |              |
| padding_word                          | 16                 | bslbf        |
| }                                     |                    |              |
| <b>TableOfPlayLists()</b>             |                    |              |
| for (i=0;i<N2;i++){                   |                    |              |
| padding_word                          | 16                 | bslbf        |
| }                                     |                    |              |
| <b>MakerPrivateData()</b>             |                    |              |
| }                                     |                    |              |

FIG.5

| SYNTAX                   | NUMBER<br>OF BYTES | ABBREVIATION |
|--------------------------|--------------------|--------------|
| DVRVolume(){             |                    |              |
| <b>version_number</b>    | 8*4                | bslbf        |
| <b>length</b>            | 32                 | uimsbf       |
| <b>ResumeVolume()</b>    |                    |              |
| <b>UIAppInfoVolume()</b> |                    |              |
| }                        |                    |              |

FIG.6

| SYNTAX                       | NUMBER<br>OF BYTES | ABBREVIATION |
|------------------------------|--------------------|--------------|
| UIAppInfoVolume() {          |                    |              |
| character_set                | 8                  | bslbf        |
| name_length                  | 8                  | uimsbf       |
| Volume_name                  | 8*256              | bslbf        |
| reserved                     | 15                 | bslbf        |
| Volume_protect_flag          | 1                  | bslbf        |
| PIN                          | 8*4                | bslbf        |
| ref_thumbnail_index          | 16                 | uimsbf       |
| reserved                     | 7                  | bslbf        |
| rp_info_valid_flag           | 1                  | uimsbf       |
| rp_ref_to_PlayList_file_name | 8*10               | bslbf        |
| rp_ref_to_PlayItem_id        | 16                 | uimsbf       |
| rp_time_stamp                | 32                 | uimsbf       |
| }                            |                    |              |

FIG.7

| SYNTAX  | NUMBER<br>OF BYTES | ABBREVIATION |
|---|--------------------|--------------|
| TableOfPlayLists() {                            |                    |              |
| <b>version_number</b>                           | 8*4                | bslbf        |
| <b>length</b>                                   | 32                 | uimsbf       |
| <b>number_of_PlayLists</b>                      | 16                 | uimsbf       |
| for (i=0; i< <i>number_of_PlayLists</i> ; i++){ |                    |              |
| <b>PlayList_file_name</b>                       | 8*10               | bslbf        |
| UIAppInfoPlayList2()                            |                    |              |
| }   |                    |              |
| }   |                    |              |

FIG.8

| SYNTAX                  | NUMBER<br>OF BYTES | ABBREVIATION |
|-------------------------|--------------------|--------------|
| UIAppInfoPlayList20{    |                    |              |
| character_set           | 8                  | bslbf        |
| name_length             | 8                  | uimsbf       |
| PlayList_name           | 8*256              | bslbf        |
| reserved                | 8                  | bslbf        |
| record_time_and_date    | 4*14               | bslbf        |
| reserved                | 8                  | bslbf        |
| duration                | 4*6                | bslbf        |
| valid_period            | 4*8                | bslbf        |
| maker_id                | 16                 | bslbf        |
| maker_code              | 16                 | bslbf        |
| reserved                | 11                 | bslbf        |
| playback_control_flag   | 1                  | bslbf        |
| write_protect_flag      | 1                  | bslbf        |
| is_played_flag          | 1                  | bslbf        |
| archive                 | 2                  | bslbf        |
| ref_thumbnail_index     | 16                 | uimsbf       |
| reserved                | 7                  | bslbf        |
| rp_info_valid_flag      | 1                  | uimsbf       |
| rp_ref_to_PlayItem_id   | 16                 | uimsbf       |
| rp_time_stamp           | 32                 | uimsbf       |
| reserved_for_future_use | 240                | bslbf        |
| }                       |                    |              |

FIG.9



| SYNTAX                         | NUMBER<br>OF BYTES | ABBREVIATION |
|--------------------------------|--------------------|--------------|
| zzzzz.clpi {                   |                    |              |
| STC_Info_Start_address         | 32                 | uimsbf       |
| ProgramInfo_Start_address      | 32                 | uimsbf       |
| CPI_Start_address              | 32                 | uimsbf       |
| ClipMark_Start_address         | 32                 | uimsbf       |
| MakerPrivateData_Start_address | 32                 | uimsbf       |
| reserved                       | 96                 | bslbf        |
| ClipInfo()                     |                    |              |
| for (i=0;i<N1;i++){            |                    |              |
| padding_word                   | 16                 | bslbf        |
| }                              |                    |              |
| STC_Info()                     |                    |              |
| for (i=0;i<N2;i++){            |                    |              |
| padding_word                   | 16                 | bslbf        |
| }                              |                    |              |
| ProgramInfo()                  |                    |              |
| for (i=0;i<N3;i++){            |                    |              |
| padding_word                   | 16                 | bslbf        |
| }                              |                    |              |
| CPI()                          |                    |              |
| for (i=0;i<N4;i++){            |                    |              |
| padding_word                   | 16                 | bslbf        |
| }                              |                    |              |
| ClipMark()                     |                    |              |
| for (i=0;i<N5;i++){            |                    |              |
| padding_word                   | 16                 | bslbf        |
| }                              |                    |              |
| MakerPrivateData()             |                    |              |
| }                              |                    |              |

FIG.10

| SYNTAX                                  | NUMBER<br>OF BYTES | ABBREVIATION |
|---|--------------------|--------------|
| ClipMark() {                            |                    |              |
| version_number                          | 8*4                | bslbf        |
| length                                  | 32                 | uimsbf       |
| number_of_Clip_marks                    | 16                 | uimsbf       |
| for (i=0; i<number_of_Clip_marks; i++){ |                    |              |
| reserved                                | 8                  | bslbf        |
| mark_type                               | 8                  | bslbf        |
| mark_time_stamp                         | 32                 | uimsbf       |
| STC_sequence_id                         | 8                  | uimsbf       |
| reserved                                | 24                 | bslbf        |
| character_set                           | 8                  | bslbf        |
| name_length                             | 8                  | uimsbf       |
| mark_name                               | 8*256              | bslbf        |
| ref_thumbnail_index                     | 16                 | uimsbf       |
| }                                       |                    |              |
| }                                       |                    |              |

FIG.11

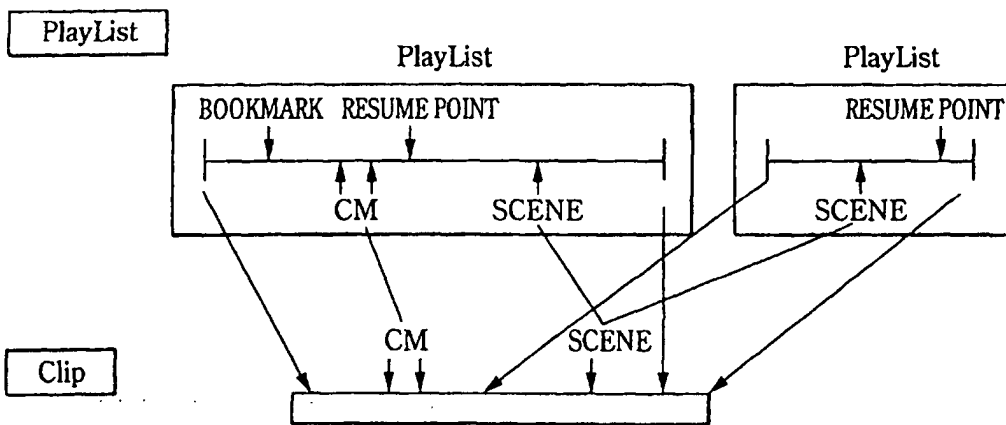


FIG.12

| SYNTAX                                | NUMBER<br>OF BYTES | ABBREVIATION |
|---------------------------------------|--------------------|--------------|
| xxxxx.rpls / yyyyy.vpls {             |                    |              |
| <b>PlayListMark_Start_address</b>     | 32                 | uimsbf       |
| <b>MakerPrivateData_Start_address</b> | 32                 | uimsbf       |
| reserved                              | 192                | bslbf        |
| <b>PlayList()</b>                     |                    |              |
| for (i=0;i<N1;i++){                   |                    |              |
| padding_word                          | 16                 | bslbf        |
| }                                     |                    |              |
| <b>PlayListMark()</b>                 |                    |              |
| for (i=0;i<N2;i++){                   |                    |              |
| padding_word                          | 16                 | bslbf        |
| }                                     |                    |              |
| <b>MakerPrivateData()</b>             |                    |              |
| }                                     |                    |              |

FIG.13

| SYNTAX                                    | NUMBER<br>OF BYTES | ABBREVIATION |
|---|--------------------|--------------|
| <b>PlayList()</b>                         |                    |              |
| <b>version_number</b>                     | 8*4                | bslbf        |
| <b>length</b>                             | 32                 | uimsbf       |
| <b>PlayList_type</b>                      | 8                  | uimsbf       |
| <b>CPI_type</b>                           | 1                  | bslbf        |
| <b>reserved</b>                           | 7                  | bslbf        |
| <b>UIAppInfoPlayList()</b>                |                    |              |
| <b>number_of_PlayItems</b> // main path   | 16                 | uimsbf       |
| if (<Virtual PlayList>){                  |                    |              |
| <b>number_of_SubPlayItems</b> // sub path | 16                 | uimsbf       |
| }else{                                    |                    |              |
| <b>reserved</b>                           | 16                 | bslbf        |
| }   |                    |              |
| for (PlayItem_id=0;                       |                    |              |
| PlayItem_id<number_of_PlayItems;          |                    |              |
| PlayItem_id++){                           |                    |              |
| <b>PlayItem()</b> //main path             |                    |              |
| }   |                    |              |
| if (<Virtual PlayList>){                  |                    |              |
| if (CPI_type==0 && PlayList_type==0){     |                    |              |
| for (i=0; i<number_of_SubPlayItems; i++)  |                    |              |
| <b>SubPlayItem()</b> //sub path           |                    |              |
| }   |                    |              |
| }   |                    |              |
| }   |                    |              |

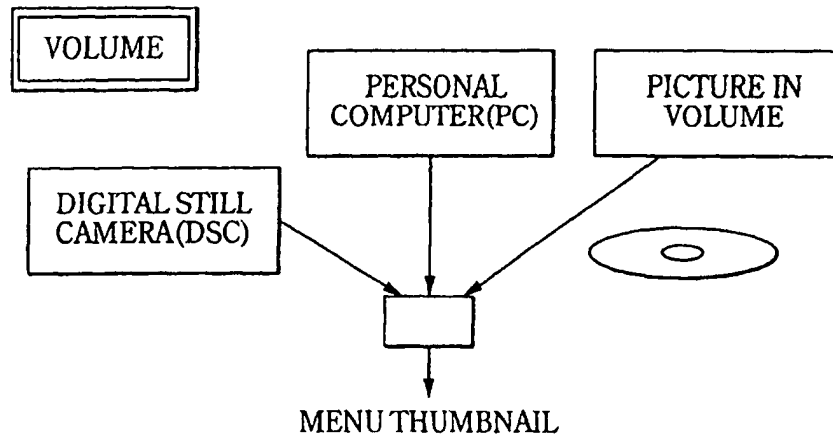
FIG.14

| SYNTAX                           | NUMBER<br>OF BYTES | ABBREVIATION |
|----------------------------------|--------------------|--------------|
| PlayItem(){                      |                    |              |
| Clip_information_file_name       | 8*10               | bslbf        |
| reserved                         | 24                 | bslbf        |
| STC_sequence_id                  | 8                  | uimsbf       |
| IN_time                          | 32                 | uimsbf       |
| OUT_time                         | 32                 | uimsbf       |
| reserved                         | 14                 | bslbf        |
| connection_condition             | 2                  | bslbf        |
| if (<Virtual PlayList>){         |                    |              |
| if (connection_condition=='10'){ |                    |              |
| BridgeSequenceInfo()             |                    |              |
| }                                |                    |              |
| }                                |                    |              |
| }                                |                    |              |

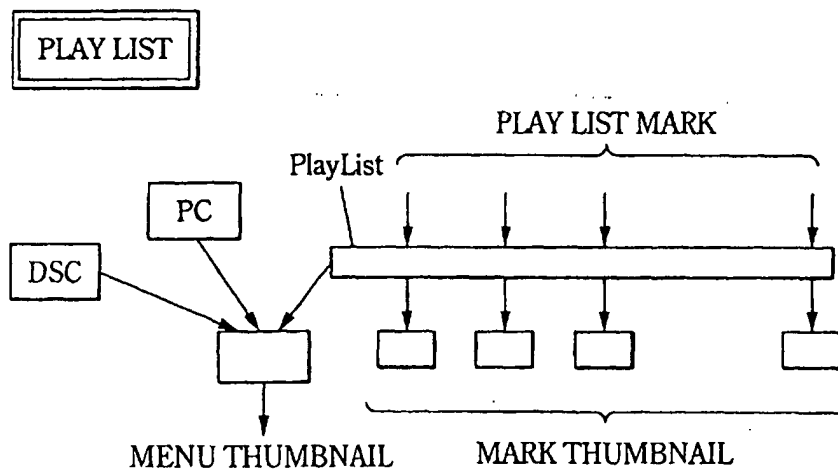
FIG.15

| SYNTAX                                    | NUMBER<br>OF BYTES | ABBREVIATION |
|---|--------------------|--------------|
| PlayListMark(){                           |                    |              |
| version_number                            | 8*4                | bslbf        |
| length                                    | 32                 | uimsbf       |
| number_of_PlayList_marks                  | 16                 | uimsbf       |
| for (i=0;i<number_of_PlayList_marks;i++){ |                    |              |
| reserved                                  | 8                  | bslbf        |
| mark_type                                 | 8                  | bslbf        |
| mark_time_stamp                           | 32                 | uimsbf       |
| PlayItem_id                               | 8                  | uimsbf       |
| reserved                                  | 24                 | uimsbf       |
| character_set                             | 8                  | bslbf        |
| name_length                               | 8                  | uimsbf       |
| mark_name                                 | 8*256              | bslbf        |
| ref_thumbnail_index                       | 16                 | uimsbf       |
| }   |                    |              |
| }   |                    |              |

FIG.16



**FIG.17**



**FIG.18**



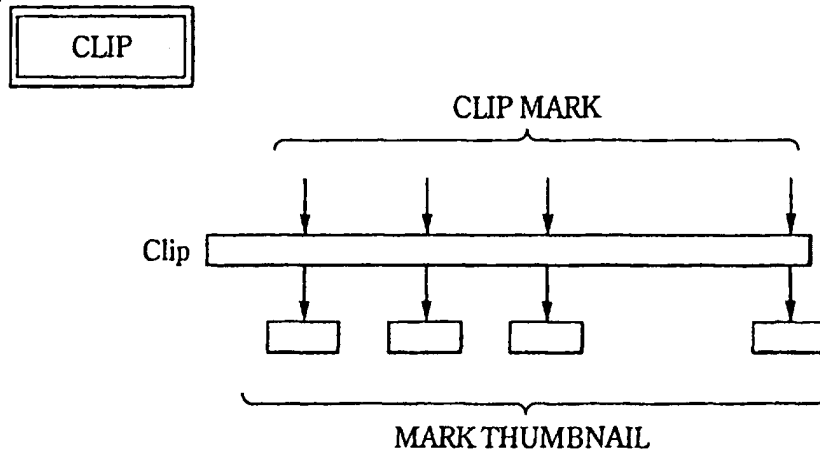


FIG.19

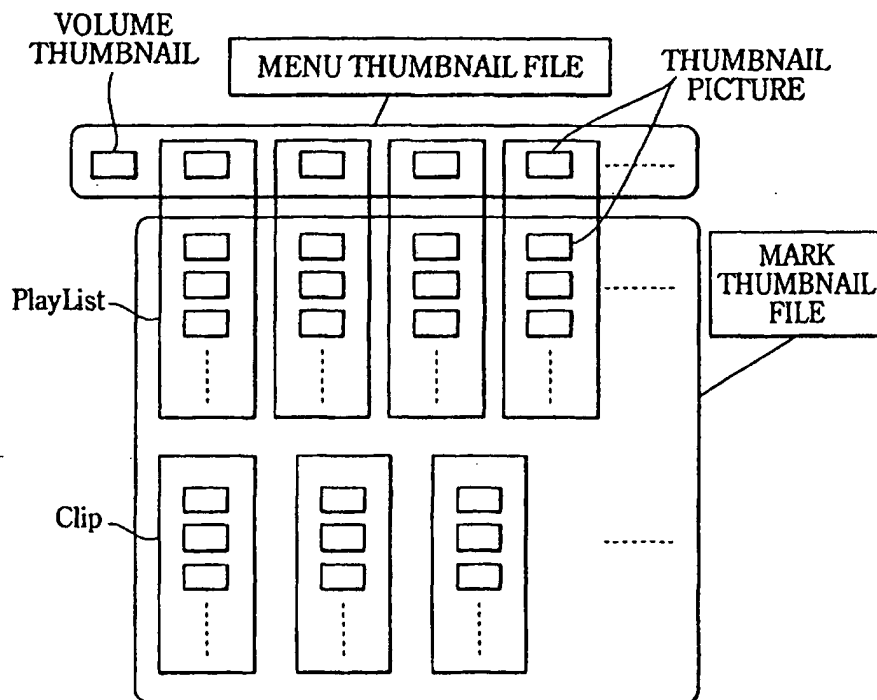


FIG.20

| SYNTAX                | NUMBER<br>OF BYTES | ABBREVIATION |
|-----------------------|--------------------|--------------|
| menu.thmb/mark.thmb { |                    |              |
| reserved              | 256                | bslbf        |
| Thumbnail()           |                    |              |
| for (i=0;i<N1;i++)    |                    |              |
| padding_word          | 16                 | bslbf        |
| }                     |                    |              |

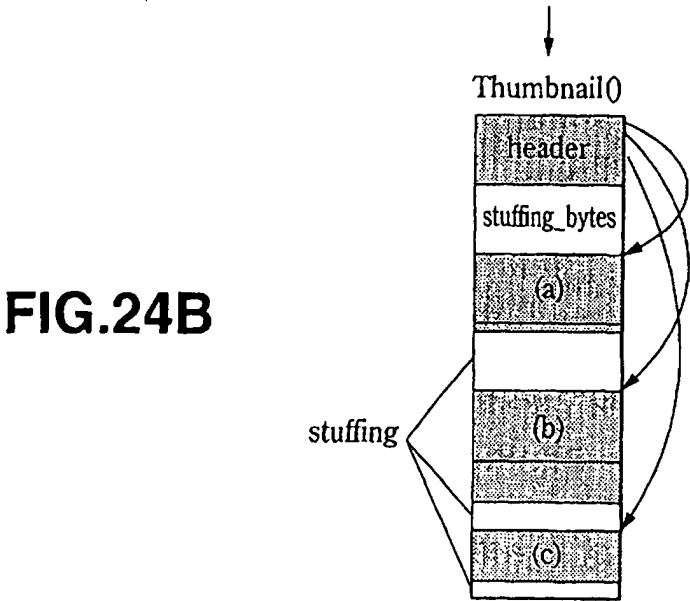
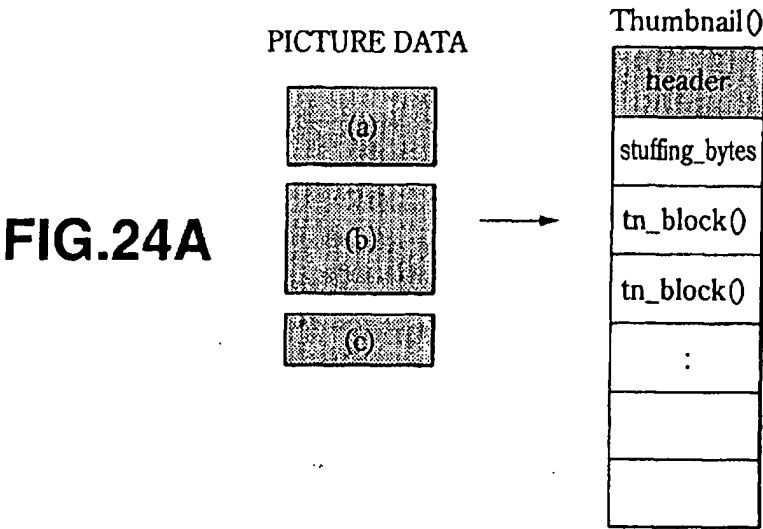
FIG.21

| SYNTAX                                   | NUMBER<br>OF BYTES | ABBREVIATION |
|--|--------------------|--------------|
| Thumbnail() {                            |                    |              |
| version_number                           | 8*4                | char         |
| length                                   | 32                 | uimsbf       |
| if (length !=0) {                        |                    |              |
| tn_blocks_start_address                  | 32                 | bslbf        |
| number_of_thumbnails                     | 16                 | uimsbf       |
| tn_block_size                            | 16                 | uimsbf       |
| number_of_tn_blocks                      | 16                 | uimsbf       |
| reserved                                 | 16                 | bslbf        |
| for (i=0; i<number_of_thumbnails; i++) { |                    |              |
| thumbnail_index                          | 16                 | uimsbf       |
| thumbnail_picture_format                 | 8                  | bslbf        |
| reserved                                 | 8                  | bslbf        |
| picture_data_size                        | 32                 | uimsbf       |
| start_tn_block_number                    | 16                 | uimsbf       |
| x_picture_length                         | 16                 | uimsbf       |
| y_picture_length                         | 16                 | uimsbf       |
| reserved                                 | 16                 | uimsbf       |
| }  |                    |              |
| stuffing_bytes                           | 8*2*L1             | bslbf        |
| for (k=0; k<number_of_tn_blocks; k++) {  |                    |              |
| tn_block                                 | fixed              |              |
| }  |                    |              |
| }  |                    |              |

FIG.22

| Thumbnail_picture_format | MEANING                |
|--------------------------|------------------------|
| 0x00                     | MPEG-2 Video I-picture |
| 0x01                     | DCF (restricted JPEG)  |
| 0x02                     | PNG                    |
| 0x03-0xff                | reserved               |

FIG.23



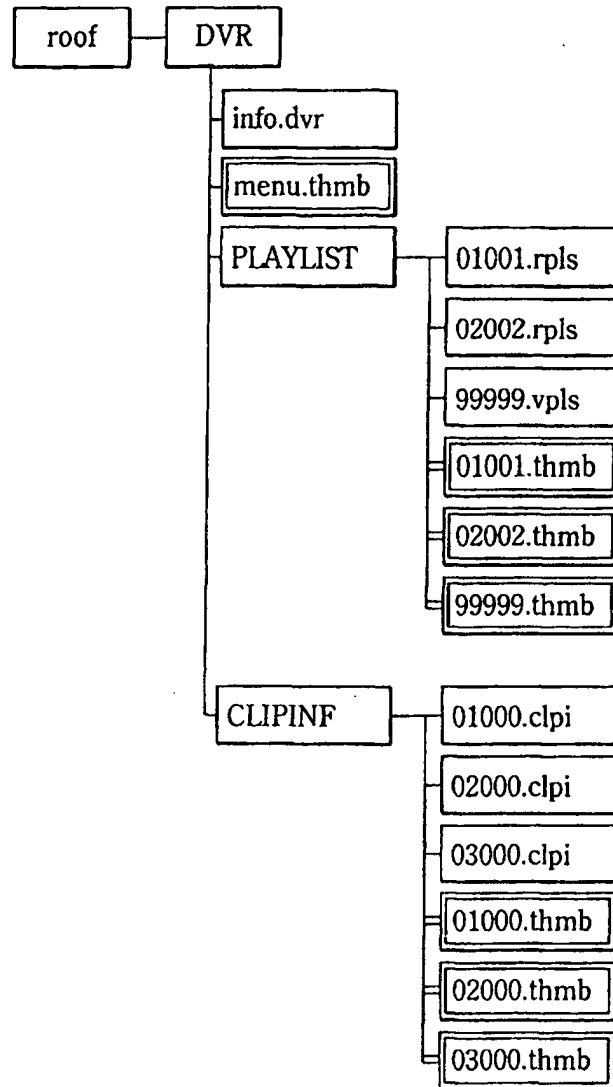


FIG.25

| SYNTAX                 | NUMBER<br>OF BYTES | ABBREVIATION |
|------------------------|--------------------|--------------|
| menu.thmb/xxxxx.thmb { |                    |              |
| reserved               | 256                | bslbf        |
| Thumbnail()            |                    |              |
| for (i=0;i<N1;i++)     |                    |              |
| padding_word           | 16                 | bslbf        |
| }                      |                    |              |

FIG.26

| SYNTAX                                 | NUMBER<br>OF BYTES | ABBREVIATION |
|--|--------------------|--------------|
| TableOfPlayLists{                      |                    |              |
| version_number                         | 8*4                | bslbf        |
| length                                 | 32                 | uimsbf       |
| number_of_PlayLists                    | 16                 | uimsbf       |
| for (i=0; i<number_of_PlayLists; i++){ |                    |              |
| PlayList_file_name                     | 8*10               | bslbf        |
| }                                      |                    |              |
| }                                      |                    |              |

FIG.27

| SYNTAX   | NUMBER<br>OF BYTES | ABBREVIATION |
|--|--------------------|--------------|
| <b>PlayList()</b> {  |                    |              |
| <b>version_number</b>  | 8*4                | bslbf        |
| <b>length</b>  | 32                 | uimsbf       |
| <b>PlayList_type</b>   | 8                  | uimsbf       |
| <b>CPI_type</b>  | 1                  | bslbf        |
| <b>reserved</b>  | 7                  | bslbf        |
| <b>UIAppInfoPlayList()</b>   |                    |              |
| <b>number_of_PlayItems</b> // main path                                    | 16                 | uimsbf       |
| if (<Virtual PlayList>){   |                    |              |
| <b>number_of_SubPlayItems</b> // sub path                                  | 16                 | uimsbf       |
| }else{   |                    |              |
| <b>reserved</b>  | 16                 | bslbf        |
| }  |                    |              |
| for ( <b>PlayItem_id</b> =0;   |                    |              |
| <b>PlayItem_id</b> < <b>number_of_PlayItems</b> ;                          |                    |              |
| <b>PlayItem_id</b> ++){  |                    |              |
| <b>PlayItem()</b> //main path  |                    |              |
| }  |                    |              |
| if (<Virtual PlayList>){   |                    |              |
| if ( <b>CPI_type</b> ==0 && <b>PlayList_type</b> ==0){                     |                    |              |
| for ( <b>i</b> =0; <b>i</b> < <b>number_of_SubPlayItems</b> ; <b>i</b> ++) |                    |              |
| <b>SubPlayItem()</b> //sub path  |                    |              |
| }  |                    |              |
| }  |                    |              |
| }  |                    |              |

FIG.28



| SYNTAX                         | NUMBER<br>OF BYTES | ABBREVIATION |
|--------------------------------|--------------------|--------------|
| UIAppInfoPlayList{             |                    |              |
| <b>character_set</b>           | 8                  | bslbf        |
| <b>name_length</b>             | 8                  | uimsbf       |
| <b>PlayList_name</b>           | 8*256              | bslbf        |
| <b>reserved</b>                | 8                  | bslbf        |
| <b>record_time_and_date</b>    | 4*14               | bslbf        |
| <b>reserved</b>                | 8                  | bslbf        |
| <b>duration</b>                | 4*6                | bslbf        |
| <b>valid_period</b>            | 4*8                | bslbf        |
| <b>maker_id</b>                | 16                 | uimsbf       |
| <b>maker_code</b>              | 16                 | uimsbf       |
| <b>reserved</b>                | 11                 | bslbf        |
| <b>playback_control_flag</b>   | 1                  | bslbf        |
| <b>write_protect_flag</b>      | 1                  | bslbf        |
| <b>is_played_flag</b>          | 1                  | bslbf        |
| <b>archive</b>                 | 2                  | bslbf        |
| <b>ref_thumbnail_index</b>     | 16                 | uimsbf       |
| <b>reserved_for_future_use</b> | 240                | bslbf        |
| }                              |                    |              |

FIG.29

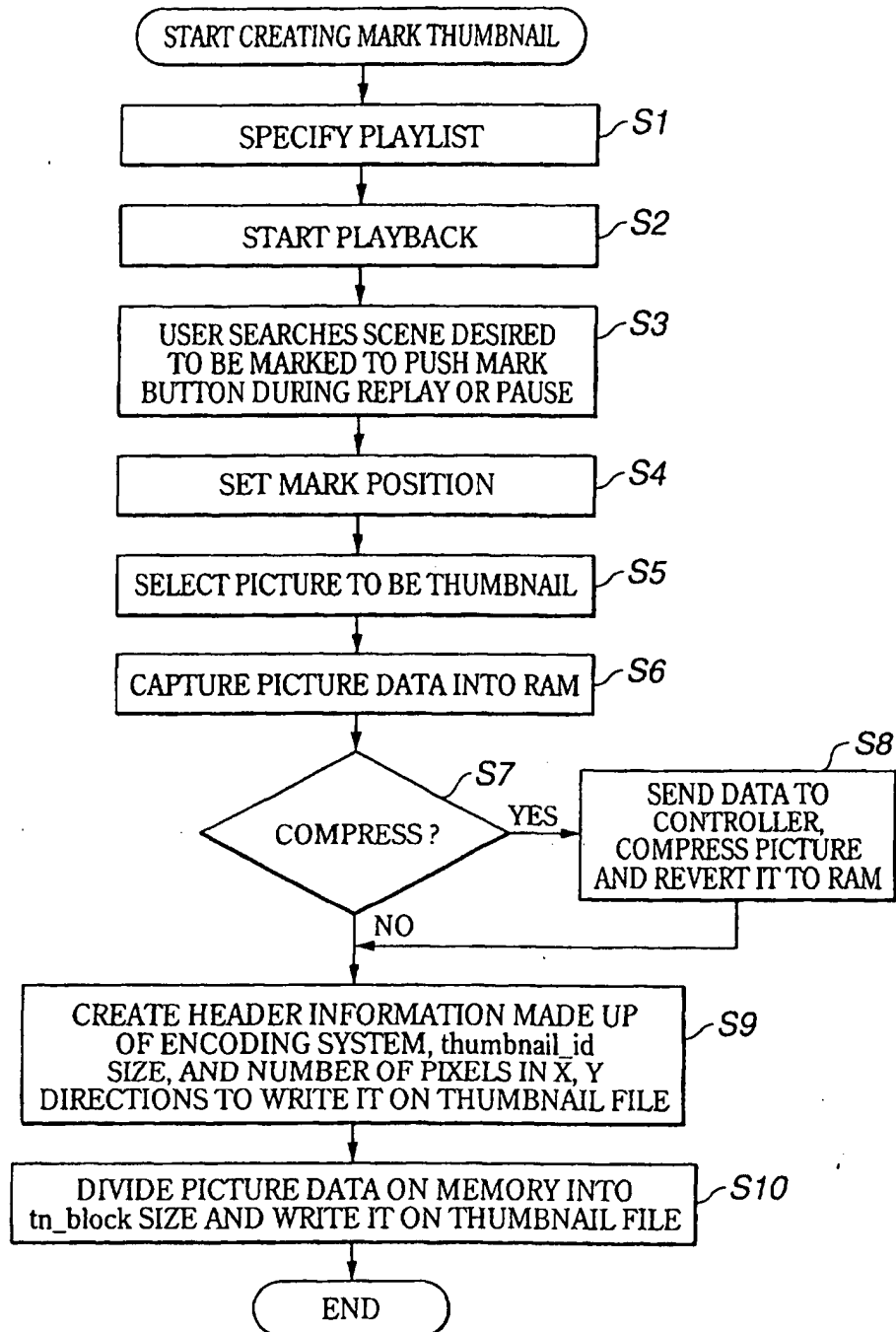


FIG.30

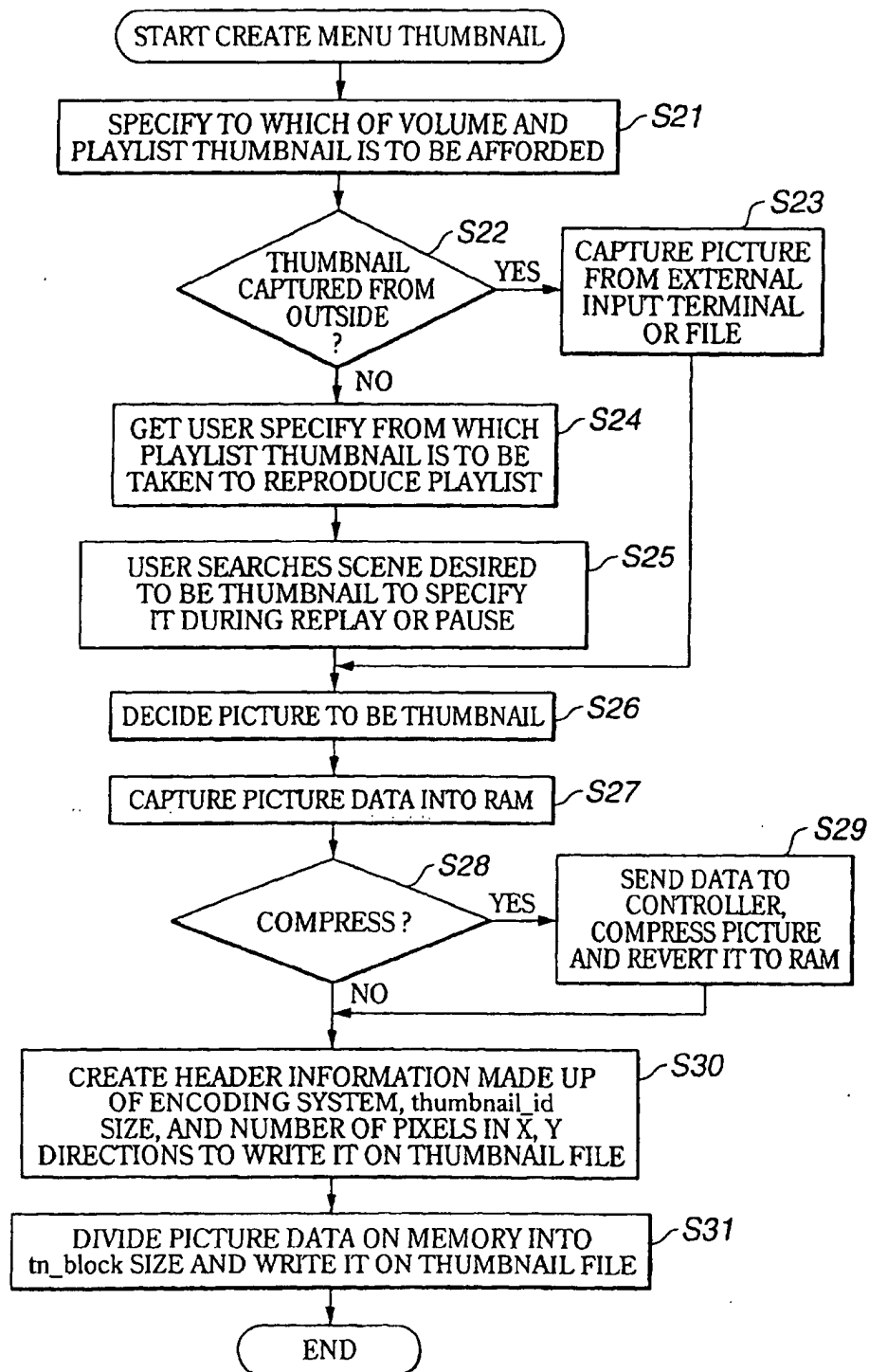
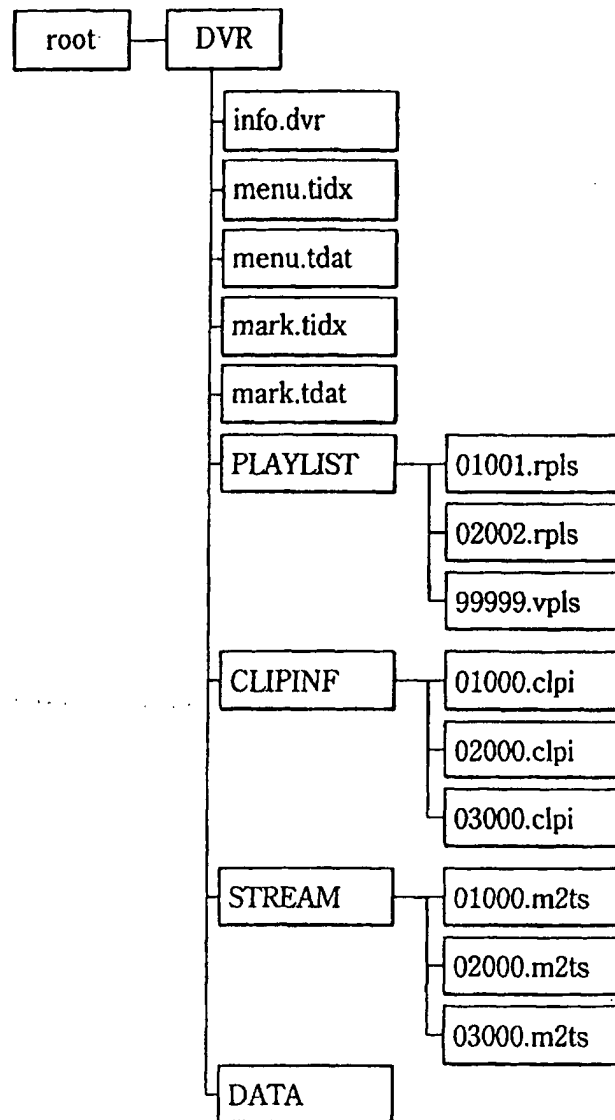


FIG.31

**FIG.32**

| SYNTAX                                  | NUMBER<br>OF BYTES | ABBREVIATION |
|---|--------------------|--------------|
| menu.tidx/mark.tidx){                   |                    |              |
| version_number                          | 8*4                | char         |
| reserved_for_future_use                 | 256                | bslbf        |
| length                                  | 32                 | uimsbf       |
| if (length !=0){                        |                    |              |
| number_of_thumbnails                    | 16                 | uimsbf       |
| tn_block_size                           | 16                 | uimsbf       |
| number_of_tn_blocks                     | 16                 | uimsbf       |
| for (i=0; i<number_of_thumbnails; i++){ |                    |              |
| thumbnail_index                         | 16                 | uimsbf       |
| ref_to_tn_block_id                      | 16                 | uimsbf       |
| picture_byte_size                       | 32                 | uimsbf       |
| horizontal_picture_size                 | 16                 | uimsbf       |
| vertical_picture_size                   | 16                 | uimsbf       |
| display_aspect_ratio                    | 4                  | uimsbf       |
| color_space                             | 4                  | uimsbf       |
| reserved_for_word_align                 | 8                  | bslbf        |
| }                                       |                    |              |
| }                                       |                    |              |
| }                                       |                    |              |

FIG.33

| display_aspect_ratio | MEANING                   |
|----------------------|---------------------------|
| 0, 1                 | reserved for future use   |
| 2                    | 4:3 display aspect ratio  |
| 3                    | 16:9 display aspect ratio |
| 4-15                 | reserved for future use   |

**FIG.34**

| display_aspect_ratio | MEANING                 |
|----------------------|-------------------------|
| 0                    | BT 709                  |
| 1                    | ITU-R Rec.601           |
| 2-15                 | reserved for future use |

**FIG.35**

| SYNTAX  | NUMBER<br>OF BYTES       | ABBREVIATION |
|---|--------------------------|--------------|
| menu.tdat/mark.tdat {   |                          |              |
| for (tn_block_id=0; tn_block_id<number_of_tn_blocks;<br>tn_block_id++){ |                          |              |
| tn_block  | tn_block_<br>size*1024*8 |              |
| }   |                          |              |
| }   |                          |              |

FIG.36

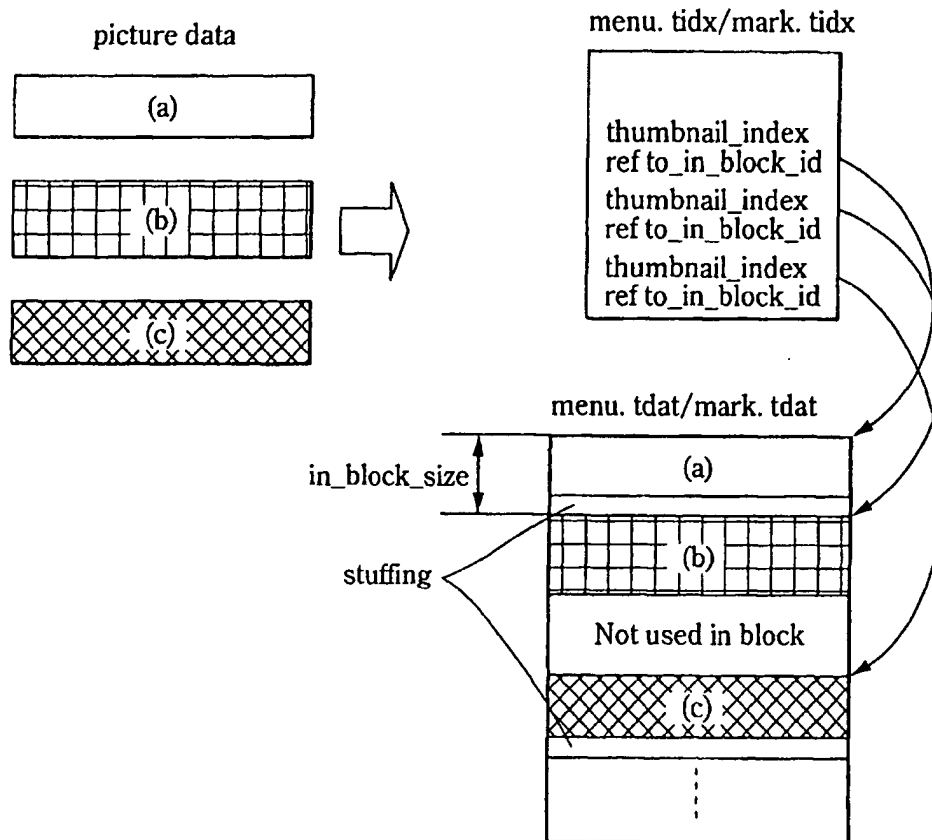
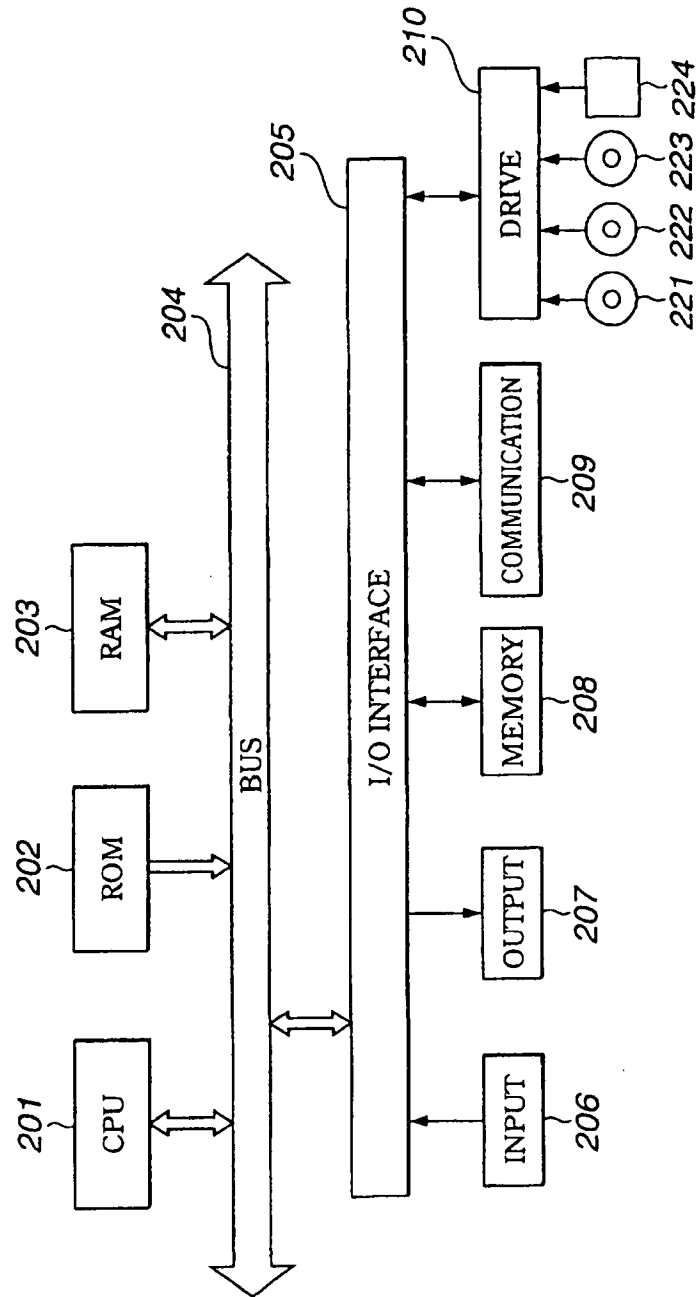


FIG.37





**FIG.38**

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/03416

| <b>A. CLASSIFICATION OF SUBJECT MATTER</b><br>Int.Cl. <sup>7</sup> H04N 5/93, G11B 20/10  |  |  |
|---|--|--|
| According to International Patent Classification (IPC) or to both national classification and IPC   |  |  |
| <b>B. FIELDS SEARCHED</b><br>Minimum documentation searched (classification system followed by classification symbols)<br>Int.Cl. <sup>7</sup> H04N 5/76-5/956, G11B 20/10-20/12  |  |  |
| Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched<br>Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2001<br>Kokai Jitsuyo Shinan Koho 1971-2001 Jitsuyo Shinan Toroku Koho 1996-2001   |  |  |
| Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  |  |  |
| <b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>   |  |  |
| Category*   | Citation of document, with indication, where appropriate, of the relevant passages                                   | Relevant to claim No.  |
| Y   | JP, 2000-083204, A (NEC Software Kobe Ltd.),<br>21 March, 2000 (21.03.00),<br>Full text; Figs. 1 to 4 (Family: none) | 1-16   |
| Y   | JP, 11-273227, A (NEC Software Kobe Ltd.),<br>08 October, 1999 (08.10.99),<br>Full text; Figs. 1 to 3 (Family: none) | 1-16   |
| A   | JP, 11-213524, A (Sony Corporation),<br>06 August, 1999 (06.08.99),<br>Full text; Figs. 1 to 12 (Family: none)       | 1-16   |
| <input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.   |  |  |
| * Special categories of cited documents:<br>"A" document defining the general state of the art which is not considered to be of particular relevance<br>"B" earlier document but published on or after the international filing date<br>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)<br>"O" document referring to an oral disclosure, use, exhibition or other means<br>"P" document published prior to the international filing date but later than the priority date claimed<br>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention<br>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone<br>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art<br>"&" document member of the same patent family |  |  |
| Date of the actual completion of the international search<br>12 June, 2001 (12.06.01)   |  | Date of mailing of the international search report<br>19 June, 2001 (19.06.01) |
| Name and mailing address of the ISA/<br>Japanese Patent Office  |  | Authorized officer   |
| Facsimile No.   |  | Telephone No.  |

Form PCT/ISA/210 (second sheet) (July 1992)